POLARIZED COMMUNITY DETECTION IN SIGNED NETWORKS VIA QUASI-ARITHMETIC MEANS

Anonymous authors
Paper under double-blind review

ABSTRACT

The 2-POLARIZED-COMMUNITIES (2PC) problem seeks two disjoint polarized communities such that most edges within each community are positive, and most edges between communities are negative. The 2PC objective encodes these requirements as the arithmetic mean of the per-node *net degree balance*. While effective, this captures only an average-driven notion of polarized structure, and may overlook other notions such as tight cores, where all selected nodes are strongly polarized, and hub-driven formations, dominated by a few high-balance nodes.

We introduce a novel one-parameter family of polarized-community objectives, where the parameter p controls a quasi-arithmetic exponential mean of net degree balances. Our objective captures the standard 2PC problem as a special case ($p \to 0$), and interpolates between core-like ($p \to -\infty$) and hub-driven ($p \to +\infty$) notions of polarized structure. We study the complexity of the resulting problems, showing **NP**-hardness in general and at the core-like extreme, a polynomial-time exact algorithm for the hub-driven extreme, and identifying tractable cases on balanced signed graphs. Algorithmically, we show that the standard 2PC peeling rule does not extend to $p \neq 0$, and design a generalized peeling algorithm that removes nodes according to the exact marginal change in the finite-p objective. Experiments on real-world datasets show that p interpolates between small, dense, highly polarized cores and larger hub-driven structures; on synthetic graphs, our methods robustly identify the ground-truth communities.

1 INTRODUCTION

Online social platforms have become major arenas for public discourse, enabling users to form alliances, voice opinions, and engage in contentious debates on topics ranging from politics to sports (Blohm et al., 2025; Amelkin et al., 2019; Zareie & Sakellariou, 2023; Abebe et al., 2021). The coexistence of *positive* (agreeable, friendly) and *negative* (hostile, antagonistic) interactions gives rise to one of the most salient phenomena in contemporary social dynamics: *polarization* (Conover et al., 2011; Hohmann et al., 2023; Garimella et al., 2017; La Cava et al., 2024). Detecting and characterizing *polarized communities*, namely groups of users who strongly agree internally yet strongly disagree with one another, is a central challenge in computational social science and network analysis. *Signed graphs*, where edges carry a positive or negative sign encoding friendship or enmity, provide the natural formal model for this setting (Harary, 1953).

Signed graphs introduce fundamental structural differences that make many algorithms designed for unsigned networks inapplicable (Derr et al., 2018; Tang et al., 2016), spurring extensive advances on signed network embeddings, signed clustering, and signed link prediction (see (Tang et al., 2016) for a comprehensive review). Among clustering approaches, most methods partition all nodes into k communities (Chiang et al., 2012; Cucuringu et al., 2019; He et al., 2022; Mercado et al., 2016; Kunegis et al., 2010), evaluating their objective over the entire graph and thereby forcing even weakly or neutrally connected nodes into some cluster. A complementary well-established formulation that directly addresses this limitation is the 2-POLARIZED-COMMUNITIES (2PC) problem, introduced by Bonchi et al. (2019) and further studied in (Tzeng et al., 2020; Gullo et al., 2024; Chen et al., 2024; Aronsson & Chehreghani, 2025; Gullo et al., 2026). Given a signed graph, 2PC seeks two disjoint communities such that most intra-community edges are positive, most inter-community edges are negative, and the number of edges satisfying these conditions is high relative to the number of selected nodes. These desiderata are jointly captured by a single objective called polarity, which assigns each selected node its *net degree balance*, i.e., the difference between its consistent and inconsistent connections, and then *aggregates these values as an average across all*

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

selected nodes. This property sets 2PC apart from alternative formulations and will be central to the generalization we pursue in this work.

This arithmetic averaging captures a specific notion of polarized structure: communities with high average net degree balance. However, average is insensitive to how net degree balance scores are distributed across the selected nodes. A high-average solution may include weakly polarized nodes, provided that they are compensated by strongly polarized members. For instance, the net degree balance sequences $(3, 3, 3, 3, 3, 3)$ and $(1, 1, 3, 3, 5, 5)$ have the same mean, yet only the former is uniformly polarized: they differ in minimum (3 vs. 1) and maximum (3 vs. 5) node-wise balance, which the mean cannot distinguish. A complementary notion is instead a *tight core*, where every selected node has high net degree balance with respect to the partition. These structures emphasize different aspects of polarization: averaging rewards overall polarity, optimizing the minimum node-wise net degree balance enforces uniformly polarized membership, and optimizing the maximum node-wise net degree balance emphasizes high-balance nodes, thus spanning a spectrum from core-like to hub-driven polarized structures. The 2PC literature has so far explored only the average case in this landscape, and the question of whether a theoretically principled family of objectives spanning this spectrum exists has remained open.

We address this gap and introduce the GENERALIZED MEAN 2-POLARIZED COMMUNITIES problem, referred to as P-MEAN-2PC. The core idea is to replace the arithmetic-mean formulation of the polarity objective (Gullo et al., 2026) with a tunable family of means parameterized by a single real exponent p. This places the alternative notions of polarized structure on a common, interpretable axis, along which p determines how the distribution of node-level balances is aggregated. Because per-node net degree balance scores can be negative, standard power means are not suitable: even exponents suppress sign information and could spuriously reward negatively polarized nodes. We instead adopt quasi-arithmetic means induced by the exponential function, which handle signed inputs naturally. As $p \to 0$, the mean reduces to the arithmetic mean, recovering standard 2PC. As $p \to -\infty$, it concentrates on the least polarized node, favoring tight communities where every member is strongly and consistently polarized. As $p \to +\infty$, it concentrates on the most polarized nodes, producing hub-driven structures. The full spectrum of intermediate values reveals qualitatively distinct polarized structures that no single fixed-p method can simultaneously capture. In an opinion network, for instance, where edges denote support and dissent among users, $p \to -\infty$ selects opposing groups whose members are all consistently embedded in the support–dissent structure, i.e., natural candidates as *opinion leaders* (Song et al., 2007); $p \to 0$ maps the discussion into two broad camps polarized on average, including more peripheral participants; and $p \to +\infty$ can identify a polarizing public figure widely supported by one side and opposed by the other.

Contributions. Our main contributions are as follows.

- We introduce P-MEAN-2PC, a generalization of 2PC based on quasi-arithmetic exponential means of per-node signed degree balance scores. The parameter p interpolates between qualitatively distinct notions of polarized structure, from core-like to hub-driven communities.
- We study the complexity of P-MEAN-2PC across key parameter regimes. The problem is **NP**-hard in general and also at the minimum-mean extreme ($p \to -\infty$), while the maximum-mean variant ($p \to +\infty$) admits a polynomial-time exact solution. On balanced signed graphs, we prove polynomial-time solvability for $p \to \{-\infty, 0\}$ via reductions to MAXCORE (Seidman, 1983) and DENSEST SUBGRAPH (Goldberg, 1984), respectively.
- We show that the peeling strategy of (Gullo et al., 2026) does not extend to $p \neq 0$: the exponential objective introduces non-local removal costs absent at $p \to 0$, as removing a node propagates objective changes to its neighbors. We then design GenPeel-p-2PC, a generalized peeling algorithm that removes nodes according to their exact marginal objective change. We also propose two ad-hoc algorithms for the limit cases $p \to -\infty$ and $p \to +\infty$.
- We evaluate our algorithms on real-world and synthetic signed networks. The results show that p acts as a structural knob, moving from small, dense, highly polarized cores to larger hub-driven formations, while maintaining robust recovery of ground-truth communities across p values.

To the best of our knowledge, no prior work has studied a theoretically grounded, single-parameter family of polarization objectives for signed graphs with complexity analysis and dedicated algorithms. Due to space constraints, full proofs are given in the ***Appendix***.

2 RELATED WORK

Polarized community detection. Bonchi et al. (2019) introduced 2PC, proved **NP**-hardness, and showed that the continuous relaxation admits an exact solution via the leading eigenvector of the signed adjacency matrix; they exploit this to derive two spectral algorithms. Chen et al. (2024)

study balanced subgraph extraction with tunable violation tolerance and adapt their region-based heuristic to the 2PC problem. Gullo et al. (2024) introduce a neural framework that learns node representations to assign nodes to polarized communities or neutrality. Aronsson & Chehreghani (2025) propose a local-search algorithm for a variant of 2PC that fosters the two communities to be balanced in size. Several other works target related signed-subgraph patterns under objectives different from polarity, including maximum balanced subgraphs (Ordozgoiti et al., 2020), maximum balanced cliques (Yao et al., 2022), local spectral polarization (Xiao et al., 2020), and dichotomy (Niu & Sariyüce, 2023); we discuss these and signed-clustering methods (Chiang et al., 2012; Cucuringu et al., 2019; He et al., 2022; Kunegis et al., 2010; Mercado et al., 2016) in ***Appendix*** D.2. Most directly related to our work, Gullo et al. (2026) establish a link between 2PC and the DENSEST SUBGRAPH (DS) problem, showing that polarity equals the arithmetic mean of per-node net degree balance scores, and exploit this to design Greedy-2PC: a peeling algorithm for 2PC that iteratively removes the node with the lowest net degree balance. Our GenPeel-p-2PC generalizes Greedy-2PC, recovering it as the $p = 0$ special case. This generalization is non-trivial: the net-degree-balance removal criterion of Greedy-2PC does not extend to $p \neq 0$, since removing a node under an exponential mean non-linearly changes the objective contribution of its neighbors, unlike the $p = 0$ case. We prove that the resulting naive greedy strategy fails for $p \neq 0$, and design GenPeel-p-2PC to compute exact marginal objective changes at each removal step.

Dense(st) subgraph discovery and generalized means. A parallel line of work in dense subgraph discovery on unsigned graphs (Lee et al., 2010; Fang et al., 2022; Lanciano et al., 2024) provides the inspiration for our generalization. The DS (Goldberg, 1984) seeks a subgraph with maximum average induced degree; the MAXCORE problem (Seidman, 1983) seeks a subgraph maximizing the minimum induced degree. Both can be effectively solved by the *peeling algorithm* (Charikar, 2003), which iteratively removes the node with smallest induced degree in the current subgraph. Veldt et al. (2021) unify DS and MAXCORE within a single parameterized objective based on the *generalized mean*, i.e., a power mean with real exponent p, applied to the node degrees in the current subgraph. They show polynomial-time solvability for any $p \geq 1$ and $p \to -\infty$ and extend the peeling paradigm to all values of p. Later, Chandrasekaran et al. (2024) show that the problem is $\mathbf{NP}$-hard for every $p \in (-\infty, 1)$. Since node degrees are non-negative, the generalized-mean objective is well-defined. In our setting, per-node net degree balance scores may be negative, making the generalized mean ill-defined (even exponents would suppress sign information); we therefore adopt quasi-arithmetic exponential means instead (Section 3). Beyond the choice of mean, the signed setting introduces further differences: unlike the unsigned case, removing a node from a solution may increase or decrease a neighbor's net degree balance; moreover, the minimum-mean extreme ($p \to -\infty$) is $\mathbf{NP}$-hard in P-MEAN-2PC (Section 4), in contrast to the framework of (Veldt et al., 2021).

3 PRELIMINARIES

Signed graphs and the 2PC problem. A *signed graph* $G = (V, E^+, E^-)$ has a node set $V = \{1, 2, ..., |V|\}$, positive edges E^+ and negative edges E^-. Its *signed adjacency matrix* $\mathbf{A} \in \{-1, 0, 1\}^{|V|\times|V|}$ has $\mathbf{A}[u, v] = 1$ if $(u, v) \in E^+$, $\mathbf{A}[u, v] = -1$ if $(u, v) \in E^-$, and 0 otherwise; $d^+(u)$ and $d^-(u)$ denote the positive and negative degree of u. A pair of communities $S_{1,2} = \{S_1, S_2\}$ consists of two disjoint subsets of V; we call $S_{1,2}$ *polarized* when most intra-community edges are positive and most inter-community edges are negative. Nodes in $V \setminus (S_1 \cup S_2)$ form the *neutral set* S_0. We denote by $\mathcal{F}(G) = \{\{S_1, S_2\} \mid S_1, S_2 \subseteq V, S_1 \cap S_2 = \emptyset, S_1 \cup S_2 \neq \emptyset\}$ the set of *feasible solutions* of G, i.e., the pairs of disjoint communities that are not both empty. A solution $S_{1,2}$ is equivalently encoded by a vector $\mathbf{x}_{S_{1,2}} \in \{-1, 0, 1\}^{|V|}$, with $\mathbf{x}_{S_{1,2}}[u] = 1$ if $u \in S_1$, $\mathbf{x}_{S_{1,2}}[u] = -1$ if $u \in S_2$, and $\mathbf{x}_{S_{1,2}}[u] = 0$ if $u \in S_0$. The 2PC problem (Bonchi et al., 2019) maximizes *polarity* over all pairs $S_{1,2}$:

$$p(S_{1,2}, G) = \frac{\mathbf{x}_{S_{1,2}}^\top \mathbf{A}\, \mathbf{x}_{S_{1,2}}}{\mathbf{x}_{S_{1,2}}^\top \mathbf{x}_{S_{1,2}}}. \tag{1}$$

2PC is $\mathbf{NP}$-hard; its continuous relaxation (2PC-RELAXED) is solvable in polynomial time via the leading eigenvector of $\mathbf{A}$ (Bonchi et al., 2019), a fact we exploit to seed our algorithm (Section 5).

Net degree balance. For a pair $S_{1,2} = \{S_1, S_2\}$, an edge (u, v) with both endpoints in $S_1 \cup S_2$ is *consistent* if it is a positive intra-community edge or a negative inter-community edge; otherwise it is *inconsistent*. The *consistent degree* $d^+_{S_{1,2}}(u)$ and *inconsistent degree* $d^-_{S_{1,2}}(u)$ of $u \in S_1 \cup S_2$ count its consistent and inconsistent incident edges (edges to S_0 are ignored); $N^+_{S_{1,2}}(u)$ and $N^-_{S_{1,2}}(u)$ denote the corresponding neighbor sets. The *net degree balance* of u w.r.t. $S_{1,2}$ is $d^\pm_{S_{1,2}}(u) =$

$d^+_{S_{1,2}}(u) - d^-_{S_{1,2}}(u)$, and $d^\pm_{S_{1,2}}(V') = \sum_{u \in V'} d^\pm_{S_{1,2}}(u)$ for any $V' \subseteq S_1 \cup S_2$. As shown in (Gullo et al., 2026), polarity can be rewritten as:

$$p(S_{1,2}, G) = \frac{1}{|S_1 \cup S_2|} \sum_{u \in S_1 \cup S_2} d^\pm_{S_{1,2}}(u), \tag{2}$$

i.e., the *arithmetic mean* of per-node net degree balance scores, a fact that drives our generalization.

Quasi-arithmetic means. To replace the arithmetic mean in Eq. 2 with a tunable family, we need means defined on sequences $\mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{R}^n$ that may contain *negative* values. Standard power means $M_p(\mathbf{a}) = \left(\frac{1}{n}\sum_i a_i^p\right)^{1/p}$ require positive inputs: even exponents suppress sign information and could spuriously reward negatively polarized nodes. We instead adopt the *quasi-arithmetic mean* induced by the exponential function (Kolmogorov–Nagumo mean):

$$E_p(\mathbf{a}) = \frac{1}{p} \log\left(\frac{1}{n}\sum_{i=1}^{n} e^{p\,a_i}\right), \quad p \neq 0. \tag{3}$$

Taking limits gives $E_{-\infty}(\mathbf{a}) = \min_i a_i$, $E_{+\infty}(\mathbf{a}) = \max_i a_i$, and $E_0(\mathbf{a}) = \frac{1}{n}\sum_i a_i$ (arithmetic mean), so E_p interpolates continuously from the minimum to the maximum as p ranges over $[-\infty, +\infty]$. Additional details about the adopted quasi-arithmetic mean can be found in ***Appendix*** A.

4 Problem Definition and Complexity Results

In this section, we define our generalized objective, the corresponding optimization problem, and our main complexity results. We begin by instantiating the quasi-arithmetic mean of Eq. 3 on the per-node net degree balance sequence of a pair of communities.

Definition 1 (p-polarity)**.** *Let $G = (V, E^+, E^-)$ be a signed graph, $S_{1,2} = \{S_1, S_2\} \in \mathcal{F}(G)$ a feasible solution, and $p \neq 0$ be a real value, the p-polarity of $S_{1,2}$ is*

$$E_p(S_{1,2}) \;=\; \frac{1}{p} \log\left(\frac{1}{|S_1 \cup S_2|} \sum_{u \in S_1 \cup S_2} e^{\,p \cdot d^\pm_{S_{1,2}}(u)}\right). \tag{4}$$

At $p = 0$, $E_p(S_{1,2})$ is defined by taking the limit as $p \to 0$, namely $E_0(S_{1,2}) = \lim_{p\to 0} E_p(S_{1,2}) = \frac{1}{|S_1 \cup S_2|}\sum_{u \in S_1 \cup S_2} d^\pm_{S_{1,2}}(u)$, which exactly recovers the standard polarity objective (cf. Eq. 2).

We consider the following optimization problem, dubbed Generalized Mean 2-Polarized Communities problem (p-mean-2pc, in short).

Problem 1 (p-mean-2pc)**.** *Given a signed graph $G = (V, E^+, E^-)$ and a real number p, find $S^*_{1,2} = \{S^*_1, S^*_2\}$, with $S^*_1, S^*_2 \subseteq V$ and $S^*_1 \cap S^*_2 = \emptyset$, maximizing the p-polarity:*

$$S^*_{1,2} = \operatorname*{arg\,max}_{S_{1,2} \in \mathcal{F}(G)} E_p(S_{1,2}) \tag{5}$$

The definition of our problem implies the following equivalence result.

Lemma 1. *The standard* 2pc *problem is equivalent to the* p-mean-2pc *problem when $p \to 0$.*

Since 2pc is **NP**-hard (Bonchi et al., 2019), the following result follows.

Theorem 1. p-mean-2pc *is* **NP***-hard.*

The two extreme $p = +\infty$ and $p = -\infty$ are obtained as $E_{+\infty}(S_{1,2}) = \lim_{p\to+\infty} E_p(S_{1,2}) = \max_{u \in S_1 \cup S_2} d^\pm_{S_{1,2}}(u)$ and $E_{-\infty}(S_{1,2}) = \lim_{p\to-\infty} E_p(S_{1,2}) = \min_{u \in S_1 \cup S_2} d^\pm_{S_{1,2}}(u)$. We refer to the corresponding optimization problems, stated next, as Max-2pc and Core-2pc, respectively.

Problem 2 (Max-2pc)**.** *Given a signed graph $G = (V, E^+, E^-)$, find $S^*_{1,2} = \{S^*_1, S^*_2\}$ with*

$$S^*_{1,2} = \operatorname*{arg\,max}_{S_{1,2} \in \mathcal{F}(G)} \max_{u \in S_1 \cup S_2} d^\pm_{S_{1,2}}(u).$$

Problem 3 (Core-2pc)**.** *Given a signed graph $G = (V, E^+, E^-)$, find $S^*_{1,2} = \{S^*_1, S^*_2\}$ with*

$$S^*_{1,2} = \operatorname*{arg\,max}_{S_{1,2} \in \mathcal{F}(G)} \min_{u \in S_1 \cup S_2} d^\pm_{S_{1,2}}(u).$$

The following results show that these problems have sharply different computational complexities.

Theorem 2. Max-2pc *can be solved in polynomial time.*

Theorem 3. Core-2pc *is* **NP***-hard.*

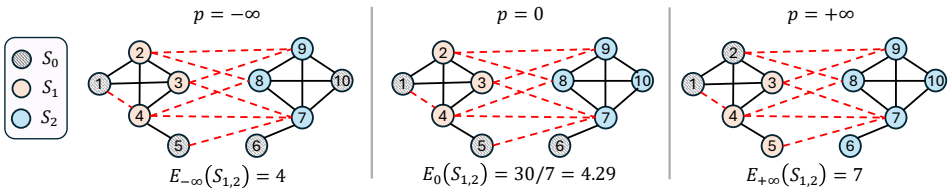


Figure 1: Optimal polarized-community pairs $S_{1,2} = \{S_1, S_2\}$ (orange and blue nodes, respectively) of P-MEAN-2PC for three representative regimes of the parameter p on the same signed graph. Solid lines and red dashed lines correspond to positive and negative edges, respectively.

The proof of Theorem 2 builds an optimal solution explicitly: it suffices to pick the node w of maximum unsigned degree and place its positive and negative neighbors in S_1 and S_2, respectively. The simplicity is specific to this extreme, as the symmetric one, $p \to -\infty$, is instead **NP**-hard.

Our CORE-2PC objective is related to the classical MAXCORE problem, i.e., maximizing the minimum induced degree in an (unsigned) subgraph. Unlike MAXCORE, however, net degree balance is computed on a signed graph and depends on the assignment to (S_1, S_2), thus capturing polarization-specific cohesion rather than plain structural density. The relation of CORE-2PC to core models for signed networks is discussed in ***Appendix*** D.2.

Varying p spans qualitatively different notions of polarized structure, from small, highly cohesive cores at $p \to -\infty$ to hub-driven formations at $p \to +\infty$. Accordingly, p is a modeling choice rather than a hyperparameter to be tuned on a given network: it specifies which notion of polarized community is sought, much as different objective functions encode different optimization goals; ***Appendix*** J discusses how to make the effect of p comparable across graphs of different scale.

Figure 1 illustrates these regimes on an example signed graph. When $p \to -\infty$, the optimum consists of a tightly cohesive pair $S_{1,2} = \{S_1, S_2\}$ with minimum net degree balance 4. At $p \to 0$, the optimum includes node 10 in S_2, which lowers the minimum net degree balance to 3 but increases the average to $30/7 \approx 4.29$. As $p \to +\infty$, the objective is dominated by the node with the highest net degree balance; the resulting solution is centered on node 7, the node with the highest unsigned degree, with its positive neighbors placed in the same community and its negative neighbors in the opposite one, thus yielding a more hub-driven polarized structure.

Polynomial cases on balanced signed graphs. We next identify polynomial-time solvable cases of P-MEAN-2PC. The key observation is that, on *balanced* signed graphs, the objective of P-MEAN-2PC can be optimized by looking only at the underlying unsigned graph, i.e. by discarding the edge signs, reducing to existing polynomial-time subgraph discovery problems on unsigned graphs.

A signed graph $G = (V, E^+, E^-)$ is balanced (Harary, 1953) if there exists a bipartition (V_1, V_2) of V (i.e., $V_1 \subseteq V$, $V_2 \subseteq V$ with $V_1 \cap V_2 = \emptyset$ and $V_1 \cup V_2 = V$) such that $E^+ \subseteq V_1 \times V_1 \cup V_2 \times V_2$ and $E^- \subseteq V_1 \times V_2 \cup V_2 \times V_1$. Such a bipartition, or a certificate that none exists, can be computed in linear time by BFS/DFS propagation of these constraints over each connected component. Connected balanced graphs admit exactly one of such bipartition.

Proposition 1. *Let $G = (V, E^+, E^-)$ be a balanced signed graph. Then, for every optimal solution $\{S_1^*, S_2^*\}$ for the* P-MEAN-2PC *problem there exists a bipartition (V_1, V_2) of V such that $S_1^* \subseteq V_1$ and $S_2^* \subseteq V_2$.*

Hence, each optimal community can be taken within one side of a balanced bipartition.

Theorem 4. *Let $G = (V, E^+, E^-)$ be a balanced signed graph. For $p \in \{-\infty, 0\}$,* P-MEAN-2PC *can be solved in polynomial time.*

The aforementioned result follows from the fact that, for $p \in \{-\infty, 0\}$, P-MEAN-2PC reduces to polynomially solvable problems (i.e., MAXCORE and DS, respectively), whereas for other values of p no such guarantee is currently available on general graphs. Although the objective is supermodular on balanced graphs for $p > 0$ (proof in ***Appendix*** C), the polynomial-time approach of (Veldt et al., 2021) does not directly apply: its binary search over a parametric threshold requires comparing objective values to determine the search direction, but our numerator objective is real-valued, so distinct thresholds may be exponentially close, precluding efficient termination.

The aforementioned result is intended to serve mainly as a theoretical result, identifying polynomial cases of the problem for $p \in \{-\infty, 0\}$. We next turn to more practical greedy algorithms.

5 ALGORITHMS

Given the success of the greedy peeling algorithm for the MAXCORE and DS problems (Harb et al., 2022) and recently for the 2PC problem (Gullo et al., 2026), it is natural to ask whether a similar

approach can be used effectively for P-MEAN-2PC when $p \neq 0$. Our algorithms start from an initial seed solution $S_{1,2}^{\text{SEED}}$ that assigns every node to one of two communities, and iteratively remove nodes to improve the objective; the best solution encountered during this process is returned. The seed is computed via EIGEN-BIPARTITION (Algorithm 2), which discretizes the leading eigenvector of the signed adjacency matrix — the optimal solution to the continuous relaxation of 2PC (Bonchi et al., 2019). We adopt this initialization to ensure a fair comparison with prior 2PC peeling methods (i.e., Greedy-2PC (Gullo et al., 2026)); in this way, differences in performance are attributable to the new peeling rule rather than to the seed construction.

To derive such a peeling rule, it is convenient to rewrite the objective in a form that makes explicit the effect of removing a node. Since $E_p(S_{1,2}) = \frac{1}{p}\log f_p(S_{1,2})$, with $f_p(S_{1,2}) = \frac{1}{|S_1 \cup S_2|}\sum_{u \in S_1 \cup S_2} e^{p \cdot d^{\pm}_{S_{1,2}}(u)}$, and $\log$ is monotonically increasing, maximizing $E_p(S_{1,2})$ is equivalent to minimizing (resp. maximizing) $f_p(S_{1,2})$ when $p < 0$ (resp. $p > 0$). For $p > 0$, the node with minimum $d^{\pm}_{S_{1,2}}(\cdot)$ also minimizes $e^{p d^{\pm}_{S_{1,2}}(\cdot)}$; for $p < 0$, the same node maximizes $e^{p d^{\pm}_{S_{1,2}}(\cdot)}$, and removing it appears to yield the largest decrease in the numerator of f_p. Thus, simple net-degree-balance peeling seems a plausible extension of the $p = 0$ strategy (Gullo et al., 2026). However, we show that this rule can perform poorly for $p \neq 0$.

Before proceeding, we establish the following non-positive-balance pruning property for $p = 0$.

Proposition 2. *Consider the* P-MEAN-2PC *problem with $p = 0$. Let $S_{1,2} = \{S_1, S_2\}$ be a feasible solution such that $E_0(S_{1,2}) > 0$. If there exists a node $z \in S_1 \cup S_2$ with $d^{\pm}_{S_{1,2}}(z) \leq 0$, then removing z from its community in $S_{1,2}$ yields a new feasible solution $S'_{1,2}$ such that $E_0(S'_{1,2}) > E_0(S_{1,2})$.*

The property above does not extend to $p \neq 0$: removing a negative-balance node may decrease the objective. The following example shows this for $p = 2$; the same issue arises for $p < 0$, for the same structural reason—removing a node affects the net degree balance of its neighbors, and these cascading effects are not captured by simple net-degree-balance peeling.

Example 1. Consider $p = 2$ and the signed graph with $V = \{h, z, v_1, v_2, w\}$, $E^+ = \{(h,z),(h,v_1),(h,v_2),(h,w)\}$, $E^- = \{(z,v_1),(z,v_2)\}$. Let $S_{1,2} = \{S_1, \emptyset\}$ with $S_1 = V$, giving net degree balances $d^{\pm}_{S_{1,2}}(h) = 4$, $d^{\pm}_{S_{1,2}}(z) = -1$, $d^{\pm}_{S_{1,2}}(v_1) = d^{\pm}_{S_{1,2}}(v_2) = 0$, $d^{\pm}_{S_{1,2}}(w) = 1$. Since $E_2(S_{1,2}) = \frac{1}{2}\log f_2(S_{1,2})$ and $\log$ is monotone, it suffices to compare f_2 values. We get $f_2(S_{1,2}) = \frac{1}{5}(e^8 + e^{-2} + 2e^0 + e^2) = 598.1$. Removing z yields $S'_{1,2} = \{S_1 \setminus z, \emptyset\}$ with $d^{\pm}_{S'_{1,2}}(h) = 3$, $d^{\pm}_{S'_{1,2}}(v_1) = d^{\pm}_{S'_{1,2}}(v_2) = d^{\pm}_{S'_{1,2}}(w) = 1$, so $f_2(S'_{1,2}) = \frac{1}{4}(e^6 + 3e^2) = 106.4$. Since $f_2(S_{1,2}) > f_2(S'_{1,2})$, removing z with $d^{\pm}_{S_{1,2}}(z) < 0$ strictly decreases the objective. □

Generalized peeling algorithm. When $p \neq 0$, removing a node with minimum net degree balance does not greedily improve the p-polarity (cf. Equation 4). For any solution $S_{1,2}$, since removing any node $v \in S_1 \cup S_2$ changes the denominator of $f_p(S_{1,2})$ in the same way, greedy improvement requires choosing the node that minimizes the numerator decrease (when $p > 0$, since we seek to maximize $f_p(\cdot)$) or maximizes it (when $p < 0$, since we seek to minimize $f_p(\cdot)$). Importantly, the best node to remove is not necessarily $z = \arg\min_{u \in S_1 \cup S_2} e^{p \cdot d^{\pm}_{S_{1,2}}(u)}$ (or $\arg\max$ when $p < 0$): this accounts only for z's direct contribution, ignoring the effect on neighbors $N^{\pm}_{S_{1,2}}(z)$. As shown in Example 1 with $p = 2$, removing a low-balance node z can lower the balance of a high-balance neighbor, significantly affecting the objective. Given $p \neq 0$, a solution $S_{1,2}$ and node $z \in S_1 \cup S_2$, the exact decrease in the numerator of $f_p(S_{1,2})$ from removing z, denoted with $\Delta_p(z; S_{1,2})$, is:

$$\Delta_p(z; S_{1,2}) = e^{p \cdot d^{\pm}_{S_{1,2}}(z)} + \sum_{u \in N^{+}_{S_{1,2}}(z)} e^{p \cdot d^{\pm}_{S_{1,2}}(u)} - e^{p \cdot (d^{\pm}_{S_{1,2}}(u) - 1)} + \sum_{u \in N^{-}_{S_{1,2}}(z)} e^{p \cdot d^{\pm}_{S_{1,2}}(u)} - e^{p \cdot (d^{\pm}_{S_{1,2}}(u) + 1)} \quad (6)$$

The numerator before removing z is $\sum_{u \in S_1 \cup S_2} e^{p \cdot d^{\pm}_{S_{1,2}}(u)}$ and after removal becomes $\sum_{u \in S_1 \cup S_2} e^{p \cdot d^{\pm}_{S_{1,2}}(u)} - \Delta_p(z; S_{1,2})$. As stated in the following proposition, the peeling rule defined in Eq. 6 generalizes the net-degree-balance rule used for $p = 0$ (Gullo et al., 2026).

Proposition 3. *As $p \to 0^+$, it holds that* $\arg\min_{z \in S_1 \cup S_2} \Delta_p(z; S_{1,2}) = \arg\min_{z \in S_1 \cup S_2} d^{\pm}_{S_{1,2}}(z)$ *and, as $p \to 0^-$, it holds that* $\arg\max_{z \in S_1 \cup S_2} \Delta_p(z; S_{1,2}) = \arg\min_{z \in S_1 \cup S_2} d^{\pm}_{S_{1,2}}(z)$.

Algorithm 1 GenPeel-p-2PC

Input: Signed graph $G = (V, E^+, E^-)$, $p \in \mathbb{R} \setminus \{0\}$
Output: Polarized communities $\widehat{S}_{1,2} = \{\widehat{S}_1, \widehat{S}_2\}$

$S_{1,2}^{\text{SEED}} \leftarrow$ EIGEN-BIPARTITION(G) {Alg. 2}
$\widehat{S}_{1,2} \leftarrow S_{1,2}^{\text{SEED}}$, $S_{1,2} \leftarrow S_{1,2}^{\text{SEED}}$
while $|S_1 \cup S_2| > 1$ **do**
  $z = \arg\min_{v \in S_1 \cup S_2} \Big(\text{sign}(p) \cdot \Delta_p(v; S_{1,2})\Big)$ {Eq. 6}
  remove z from S_1 or S_2 in $S_{1,2}$
  if $E_p(S_{1,2}) > E_p(\widehat{S}_{1,2})$ **then**
    $\widehat{S}_{1,2} \leftarrow S_{1,2}$

Algorithm 2 EIGEN-BIPARTITION

Input: Signed graph $G = (V, E^+, E^-)$
Output: Polarized communities $S_{1,2} = \{S_1, S_2\}$

Compute $\mathbf{z}^*$, the eigenvector corresponding to the largest eigenvalue λ_1 of the signed adjacency matrix $\mathbf{A}$ of G
$S_1 = \{u \in V : \mathbf{z}_u^* \geq 0\}$, $S_2 = \{u \in V : \mathbf{z}_u^* < 0\}$

Based on these observations, we define a generalized peeling algorithm, which we call GenPeel-p-2PC, based on iteratively removing nodes that minimize Eq. 6 when $p > 0$, or maximizes it when $p < 0$, i.e. picks the node minimizing $\text{sign}(p) \cdot \Delta_p(z; S_{1,2})$. Pseudocode is given in Algorithm 1.

Proposition 4. *GenPeel-p-2PC runs in $\mathcal{O}(|V||E|)$ time.*

Algorithms for $p = \pm\infty$. For $p = +\infty$, the optimal solution is computed directly by the polynomial-time construction in the proof of Theorem 2. We refer to this algorithm as Max-2PC-Exact.

For $p = -\infty$, we do not use GenPeel-p-2PC since $\Delta_p(z; S_{1,2})$ is defined for finite $p \neq 0$ and is tied to the exponential objective, whereas the limit objective is $\min_{v \in S_1 \cup S_2} d^{\pm}_{S_{1,2}}(v)$. Instead, we adopt a dedicated peeling algorithm, dubbed Core-2PC-Peel, that iteratively removes the node with minimum current net degree balance $d^{\pm}_{S_{1,2}}(v)$ and returns the solution maximizing $\min_{v \in S_{1,2}} d^{\pm}_{S_{1,2}}(v)$ over the peeling sequence. This mirrors the classical peeling strategy for the maximum k-core in unsigned graphs (Seidman, 1983), here adapted to the signed setting by replacing induced degree with net degree balance relative to a pair of polarized communities. As in GenPeel-p-2PC, the Core-2PC-Peel algorithm is initialized with EIGEN-BIPARTITION and runs in $\mathcal{O}(|V|+|E|)$ time (pseudocode and computational complexity analysis of Core-2PC-Peel are in Appendix F).

On approximation guarantees. Establishing a practical approximation guarantee for GenPeel-p-2PC is a non-trivial open problem. Already at $p = 0$, the additive guarantee of Greedy-2PC (Gullo et al., 2026) holds under the assumption that the spectral seed assigns every node of an optimal solution to its correct side, i.e., $S_1^* \subseteq S_1^{\text{SEED}}$ and $S_2^* \subseteq S_2^{\text{SEED}}$. Extending that analysis to $p \neq 0$ is further complicated by the fact that each removal changes the exponential contribution of *all* neighbors of the removed node (cf. Eq. 6); moreover, a direct extension would retain a condition depending on the unknown optimum, making the guarantee unverifiable a priori and of limited practical applicability. A mild guarantee is nonetheless available for Max-2PC-Exact: for every finite $p > 0$, the Max-2PC-Exact algorithm is an additive $\frac{\log |V|}{p}$-approximation for P-MEAN-2PC (see Proposition 5 in ***Appendix*** B). The bound is informative only for large p, while deriving guarantees for GenPeel-p-2PC or Core-2PC-Peel remains open.

6 EXPERIMENTS

Datasets. We utilized publicly available *real-world* signed graphs that are commonly adopted in prior work on polarized community detection (Aronsson & Chehreghani, 2025; Gullo et al., 2026; Tzeng et al., 2020). Their key characteristics are reported in ***Appendix*** G. We also evaluated our methods on *synthetic* signed graphs to assess their ability to recover ground-truth polarized communities. We generate them with the *modified signed stochastic block model* (M-SSBM) (Tzeng et al., 2020), parameterized by the total number of nodes, the size $n_c = |S_1| = |S_2|$ of the planted polarized communities (all of equal size), and a noise parameter $\eta \in [0, 1]$ controlling edge probabilities. Details on the synthetic generation scheme are given in ***Appendix*** H.

Evaluation goals. Our experiments address five questions: (**RQ1**) How does p affect the structure and quality of discovered polarized communities on real-world networks? (**RQ2**) Does varying p produce distinct communities, or do solutions largely overlap across different values of p? (**RQ3**) On synthetic graphs, how does p affect ground-truth recovery and precision/size trade-offs under increasing noise? (**RQ4**) How does the computational cost change with p and graph size? (**RQ5**) How close are our solutions to the exact optimum on small graphs, and how do they compare with existing methods? We solve P-MEAN-2PC for $p \in \{-\infty, -0.01, -0.001, 0, 0.001, 0.01, +\infty\}$, using the state-of-the-art Greedy-2PC (Gullo et al., 2026) for p=0, Core-2PC-Peel and Max-2PC-Exact

Table 1: Results on real-world networks for selected values of p. For each dataset, we report the average net degree balance E_0, the min. and max. net degree balance in the yielded solution, the edge-agreement ratio (agr.), the solution edge density (dens.), and the community sizes $|S_1|/|S_2|$.

criteria	p	Bitc.	Cloi.	Cong.	Epin.	HTri.	Slash.	TwRef.	WCon.	WEle.	WPol.	Word
E_0	$-\infty$	28.88	7.17	6.70	140.1	**6.18**	76.39	161.7	163.7	69.53	84.86	22.00
	-0.01	30.53	7.38	6.70	167.9	5.73	82.20	170.0	183.3	71.86	88.85	24.96
	-0.001	**30.57**	7.38	6.70	171.1	5.73	**82.80**	174.5	195.4	72.77	90.01	**25.03**
	0	**30.57**	**7.45**	6.70	**171.2**	**6.18**	**82.80**	**174.7**	**196.6**	**72.79**	**90.07**	24.96
	$+0.001$	**30.57**	**7.45**	**6.74**	171.1	**6.18**	82.78	174.3	30.46	72.78	20.62	24.96
	$+0.01$	4.41	**7.45**	**6.74**	30.13	**6.18**	7.08	34.01	30.46	37.07	20.62	24.93
	$+\infty$	4.39	5.67	3.12	24.43	4.73	7.02	26.06	29.08	24.05	18.05	12.27
$\min / \max d^{\pm}_{S_{1,2}}(u)$	$-\infty$	**19**/69	**5**/9	**4**/16	**121**/148	**5**/8	**53**/121	**103**/347	**141**/314	**41**/275	**51**/230	**22**/22
	-0.01	16/88	**5**/10	**4**/16	99/221	3/9	45/166	95/420	111/544	36/347	46/326	13/66
	-0.001	16/93	**5**/10	**4**/16	88/262	3/9	42/184	89/537	103/687	37/425	45/405	13/67
	0	16/93	4/10	**4**/16	86/268	5/8	42/184	88/567	102/718	37/452	46/424	13/66
	$+0.001$	16/93	4/10	**4**/14	83/273	5/8	41/188	81/627	-362/20056	36/457	-26/10390	13/66
	$+0.01$	-1/792	4/10	**4**/14	-28/3421	5/8	-6/2530	-3/2390	-362/20056	-17/909	-26/10390	12/70
	$+\infty$	-4/**795**	-1/**17**	1/**33**	-593/**3558**	1/**10**	-16/**2548**	-448/**2784**	-3277/**20146**	-185/**1065**	-708/**10715**	-1/**157**
agr.	$-\infty$	**.972**	**.939**	.995	**1.00**	**1.00**	**.995**	**.997**	**.982**	**.933**	**.971**	**1.00**
	-0.01	.962	.914	.995	**1.00**	.906	.988	**.997**	.977	.930	.970	.980
	-0.001	.963	.914	.995	**1.00**	.906	.987	.996	.974	.927	.969	.979
	0	.963	.936	.995	**1.00**	**1.00**	.987	.996	.972	.924	.969	.980
	$+0.001$	.963	.936	.995	**1.00**	**1.00**	.987	.996	.825	.924	.914	.980
	$+0.01$	.971	.936	.995	.951	**1.00**	.921	.981	.825	.869	.914	.978
	$+\infty$	.966	.704	**1.00**	.803	.906	.910	.781	.800	.700	.832	.922
dens.	$-\infty$	**.262**	.742	.109	**.947**	**.618**	**.628**	**.427**	**.529**	**.193**	**.331**	**1.00**
	-0.01	.181	.744	.109	.760	.505	.481	.346	.206	.148	.234	.234
	-0.001	.174	.744	.109	.651	.505	.423	.273	.183	.124	.186	.238
	0	.174	**.855**	.115	.637	**.618**	.423	.259	.178	.116	.176	.238
	$+0.001$	.174	**.855**	**.122**	.625	**.618**	.413	.233	.002	.115	.002	.238
	$+0.01$	.006	**.855**	**.122**	.010	**.618**	.003	.015	.002	.055	.002	.217
	$+\infty$	.006	.817	.094	.011	.582	.003	.017	.002	.056	.003	.093
$\lvert S_1\rvert/\lvert S_2\rvert$	$-\infty$	89/29	9/3	35/28	148/1	7/4	124/0	379/3	221/101	414/2	273/0	14/9
	-0.01	152/32	10/3	35/28	221/1	11/4	176/0	493/3	659/276	562/2	405/0	67/45
	-0.001	159/32	10/3	35/28	263/1	11/4	202/0	640/4	825/303	688/2	516/2	66/45
	0	159/32	8/3	34/26	269/1	7/4	202/0	676/4	865/305	740/2	546/2	65/45
	$+0.001$	159/32	8/3	30/27	274/1	7/4	207/0	750/4	18641/1416	745/2	8061/2330	65/45
	$+0.01$	783/10	8/3	30/27	3294/128	7/4	2509/22	2391/12	18641/1416	776/135	8061/2330	72/49
	$+\infty$	786/10	13/5	29/5	3324/235	6/5	2518/31	1540/1245	18715/1432	820/246	8075/2641	107/51

for the $\pm\infty$ cases respectively, and GenPeel-p-2PC (Algorithm 1) for all finite $p \neq 0$. Inspired by (Veldt et al., 2021), we focus on a narrow range around the standard 2PC ($p = 0$) regime, i.e., $p \in [-0.01, 0.01]$, where the objective exhibits the highest sensitivity to p by definition (cf. Eq.4) and the main structural changes in the resulting communities already emerge; the limit cases $p = \pm\infty$ yield only incremental further changes (cf. Table 1). For **RQ5** we further consider four representative baselines: SPONGE (Cucuringu et al., 2019), SSSNET (He et al., 2022), TIMBAL (Ordozgoiti et al., 2020), and RH (Chen et al., 2024).

Evaluation criteria. Since real-world datasets lack ground-truth communities, we follow prior work (Tzeng et al., 2020; Gullo et al., 2026) and use *polarity* (Eq. 1) as a quality proxy, together with *edge density* and the *edge-agreement ratio*, i.e., the fraction of consistent edges in the solution. Across pairs of tested values p_1, p_2, we compare yielded node sets $S^{(p_i)} = S_1^{(p_i)} \cup S_2^{(p_i)}$ via *Jaccard similarity* $J(S^{(p_1)}, S^{(p_2)}) = \frac{|S^{(p_1)} \cap S^{(p_2)}|}{|S^{(p_1)} \cup S^{(p_2)}|}$ and *containment ratio* $C(S^{(p_1)}, S^{(p_2)}) = \frac{|S^{(p_1)} \cap S^{(p_2)}|}{|S^{(p_1)}|}$, capturing symmetric overlap and asymmetric containment, respectively. For datasets with available ground-truth, we measure the recovery rate of ground-truth clusters using the F_1-*score*, which is the precision and recall averaged over all clusters (as in (Tzeng et al., 2020; Gullo et al., 2026)).

Results. Table 1 reports results on 11 real-world signed networks, addressing **RQ1**. Overall, p acts as a structural-quality knob: $p \to -\infty$ and small negative values favor dense, high-agreement solutions in which every selected node has positive net degree balance, whereas positive values expand the solutions around high-balance nodes at the expense of density, agreement and minimum balance. Averaged over the datasets, the agreement ratio drops from .98 ($p \to -\infty$) and .975 ($p = 0$) to .94 ($p = +0.01$) and .848 ($p \to +\infty$), and density from .526 to .344, .173 and .153, respectively; for $p \leq 0$ we have $\min_u d^{\pm}_{S_{1,2}}(u) > 0$ on every dataset, while $\min_u d^{\pm}_{S_{1,2}}(u) < 0$ on most datasets at $p = +0.01$ and $p \to +\infty$. Within the non-positive regime, moving from $p \to -\infty$ to $p = 0$ increases E_0 while decreasing $\min_u d^{\pm}_{S_{1,2}}(u)$ on 9 datasets; for $p > 0$, solutions expand sharply on the larger networks, growing by at least $3\times$ from $p = 0$ to $p = +0.01$ on most datasets. The hub-driven regime weakens global quality, with agreement no greater than .80 in 4 cases, but can recover two non-trivially sized communities when the non-positive ones are one-sided: on Slash. and

WPol., where $p \leq 0$ often yields an empty or nearly empty second community, $p \to +\infty$ achieves agreement .910 and .832. Small datasets are more stable: Cong., HTri., Cloi. and Word maintain high agreement across finite p and show limited size variation (***Appendix*** K provides visualizations for representative solutions on Word). Moreover, finite $p \neq 0$ can slightly improve E_0 over Greedy-2PC (i.e., on Cong. and Word), although Greedy-2PC greedily optimizes E_0.

Figure 2 reports the average Jaccard similarity over the 11 real-world datasets (**RQ2**); the corresponding containment ratios and the per-dataset heatmaps are in ***Appendix*** K. Solutions for $p \in \{-0.01, -0.001, 0\}$ are highly similar, while the smaller $p \to -\infty$ solutions (cf. Table 1) are mostly contained in the solutions with $p < 0$. The main transition occurs for $p > 0$: similarity with $p = 0$ drops from .76 at $p = +0.001$ to .36 at $p = +0.01$, while $p = +0.01$ is closer to $p \to +\infty$ (Jaccard .69), indicating a high-balance-node-centered regime. The containment ratios reported in ***Appendix*** K show that nesting among solutions weakens across the transition, revealing a restructuring of the yielded communities. This effect is strongest on large sparse networks (e.g., WCon., WPol.), where overlap between non-positive and large-positive regimes is often near zero; it is softer on smaller or denser graphs such as HTri. and Cloi. (cf. ***Appendix*** K).

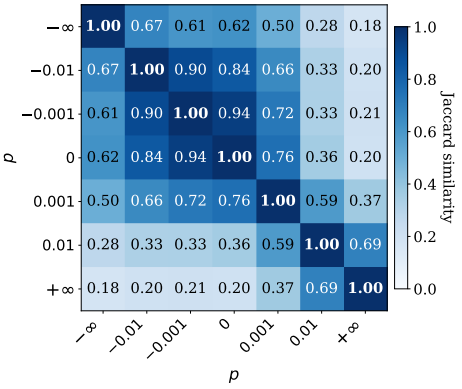


Figure 2: Average Jaccard similarity between the node sets returned by different values of p, over the 11 real-world datasets.

Concerning **RQ3**, for low and moderate noise all finite p values almost exactly recover the ground-truth communities: $F_1 = 1$ for $\eta \leq 0.3$, $F_1 \geq .997$ at $\eta = 0.4$, and $F_1 \geq .947$ at $\eta = 0.5$. The extremes exhibit different trade-offs: $p \to -\infty$ is more conservative, with lower recall but perfect precision up to $\eta = 0.5$, while $p \to +\infty$ returns larger communities but rapidly loses accuracy as noise increases. In terms of structural quality, finite p values match or slightly improve the $p = 0$ polarity, while $p \to -\infty$ gives the highest agreement for $\eta > 0$ at slightly lower polarity. Overall, finite p preserves standard 2PC recovery, whereas the extremes expose precision/size trade-offs; the corresponding plots, together with precision, recall and solution sizes, are in ***Appendix*** K.

As for runtimes (**RQ4**), Max-2PC-Exact is fastest (2.63s on average), followed by Core-2PC-Peel (6.08s), and Greedy-2PC (49.62s); full results are in ***Appendix*** K. GenPeel-p-2PC typically has higher runtimes (435–880s on average) due to exact marginal-gain computations (cf. Eq. 6), with overhead concentrated on the largest graphs, e.g., WCon. ($\sim$ 3k–7k s, vs. 124s for $p = 0$ and 13.2–22.1s for the limit cases), while runtimes on small/medium networks stay typically below 10s. Hardware details are provided in ***Appendix*** I.

Concerning **RQ5**, each of our algorithms attains the best value on the criterion it targets: Core-2PC-Peel on the minimum net degree balance (11/11 datasets) and on density (8/11), and Max-2PC-Exact on the maximum net degree balance (11/11). The best E_0 is always attained within the P-MEAN-2PC family, and only RH, the strongest baseline, ever ties it, on a single dataset. Excluding TIMBAL, whose perfect agreement follows by construction, Core-2PC-Peel also attains the highest average agreement ratio (.98, versus .969 for RH), so cohesion is not bought at the cost of agreement. Finally, on small graphs where the optimum can be computed exactly, GenPeel-p-2PC reaches 98.7%–99.9% of it, providing direct evidence of near-optimal behavior despite the absence of an approximation guarantee. Per-dataset results and the detailed analysis are in ***Appendix*** K.

7 CONCLUSION

We introduced P-MEAN-2PC, a generalization of 2PC based on quasi-arithmetic exponential means that interpolates between core-like and hub-driven polarized structures. We characterized the complexity of the problem across key regimes and designed dedicated algorithms. Experiments on real-world networks confirmed that the main parameter of our framework acts as a structural knob exposing qualitatively distinct communities, while experiments on synthetic datasets showed robust recovery of ground-truth communities. Although the proposed GenPeel-p-2PC and Core-2PC-Peel are empirically effective, they lack approximation guarantees; deriving such guarantees remains open. Moreover, we focus on two polarized communities plus a neutral set, while many settings may involve multiple antagonistic groups. Future work includes investigating approximation guarantees for our algorithms, settling the complexity of P-MEAN-2PC on balanced signed graphs for finite $p \neq 0$, and extensions to k polarized communities (Tzeng et al., 2020).

AI USE STATEMENT

In this work, we have not used generative AI tools to generate synthetic data sets, develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on the research methodology or the experiments, implement methods, clean and reformat datasets, or interpret results; assisting with translation and supporting qualitative and thematic data analysis are not applicable to this work. Additionally, we used generative AI tools to edit the paper to improve readability and to polish the wording of text drafted by the authors. We have reviewed all AI-assisted work: every edited passage was checked by the authors against the paper's definitions, proofs and experimental results, and rewritten wherever the edit altered the intended meaning. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Our study is mainly of a theoretical nature and its societal impact is indirect. On the positive side, we see two applications. First, the rise of polarization and misinformation is tied to the existence of polarized communities, and an efficient characterization tool is a first step towards mitigating it. Second, collaboration and competition coexist in most communities, and detecting conflicting groups helps understand the interplay between the two. With respect to negative consequences, we do not foresee specific issues when applying our method.

REPRODUCIBILITY STATEMENT

All theoretical claims are stated with their assumptions in the main paper and proved in ***Appendix*** B. The pseudocode of all our algorithms is given in Algorithms 1–3. All real datasets are publicly available from the cited sources; their characteristics and provenance are reported in ***Appendix*** G. The generative model used for the synthetic graphs, together with the values of its parameters and the number of instances per setting, is described in ***Appendix*** H. The hardware used for all experiments is reported in ***Appendix*** I. The source code will be made publicly available upon acceptance.

REFERENCES

Rediet Abebe, T-H Hubert Chan, Jon Kleinberg, Zhibin Liang, David Parkes, Mauro Sozio, and Charalampos E Tsourakakis. Opinion dynamics optimization by varying susceptibility to persuasion via non-convex local search. *TKDD*, 16(2):1–34, 2021.

Victor Amelkin, Petko Bogdanov, and Ambuj K Singh. A distance measure for the analysis of polar opinion dynamics in social networks. *TKDD*, 13(4):1–34, 2019.

Linus Aronsson and Morteza Haghir Chehreghani. An efficient local search approach for polarized community discovery in signed networks. In *Proc. NIPS Conf.*, 2025. URL https://openreview.net/forum?id=0E2Um9n3YK.

Peter Blohm, Florian Chen, Aristides Gionis, and Stefan Neumann. Discovering opinion intervals from conflicts in signed graphs. In *Proc. NIPS Conf.*, 2025. URL https://openreview.net/forum?id=zJdutIT6vT.

Francesco Bonchi, Edoardo Galimberti, Aristides Gionis, Bruno Ordozgoiti, and Giancarlo Ruffo. Discovering polarized communities in signed networks. In *Proc. CIKM Conf.*, pp. 961–970, 2019.

Digvijay Boob, Yu Gao, Richard Peng, Saurabh Sawlani, Charalampos Tsourakakis, Di Wang, and Junxing Wang. Flowless: Extracting densest subgraphs without flow computations. In *Proc. WWW Conf.*, pp. 573–583, 2020.

Karthekeyan Chandrasekaran, Chandra Chekuri, Manuel R Torres, and Weihao Zhu. On the generalized mean densest subgraph problem: Complexity and algorithms. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2024)*, pp. 9–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2024.

Moses Charikar. Greedy approximation algorithms for finding dense components in a graph. In *Proc. APPROX Work.*, pp. 84–95, 2003.

Chandra Chekuri, Kent Quanrud, and Manuel R Torres. Densest subgraph: Supermodularity, iterative peeling, and flow. In *Proc. SODA Conf.*, pp. 1531–1555, 2022.

Jingbang Chen, Qiuyang Mang, Hangrui Zhou, Richard Peng, Yu Gao, and Chenhao Ma. Scalable algorithm for finding balanced subgraphs with tolerance in signed networks. In *Proc. KDD Conf.*, pp. 278–287, 2024.

Kai-Yang Chiang, Joyce Jiyoung Whang, and Inderjit S Dhillon. Scalable clustering of signed networks using balance normalized cut. In *Proc. CIKM Conf.*, pp. 615–624, 2012.

Lingyang Chu, Zhefeng Wang, Jian Pei, Jiannan Wang, Zijin Zhao, and Enhong Chen. Finding gangs in war from signed networks. In *Proc. KDD Conf.*, pp. 1505–1514, 2016.

M. Conover, J. Ratkiewicz, M. Francisco, B. Goncalves, F. Menczer, and A. Flammini. Political polarization on Twitter. In *Proc. ICWSM Conf.*, pp. 89–96, 2011.

Mihai Cucuringu, Peter Davies, Aldo Glielmo, and Hemant Tyagi. SPONGE: A generalized eigenproblem for clustering signed networks. In *Proc. AISTATS Conf.*, pp. 1088–1098, 2019.

Tyler Derr, Yao Ma, and Jiliang Tang. Signed graph convolutional networks. In *Proc. ICDM Conf.*, pp. 929–934, 2018.

Yixiang Fang, Wensheng Luo, and Chenhao Ma. Densest subgraph discovery on large graphs: Applications, challenges, and techniques. *Proc. VLDB Conf.*, 15(12):3766–3769, 2022.

Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. Reducing controversy by connecting opposing views. In *Proc. WSDM Conf.*, pp. 81–90, 2017.

Michael U Gerber and Daniel Kobler. Algorithmic approach to the satisfactory graph partitioning problem. *European Journal of Operational Research*, 125(2):283–291, 2000.

Andrew V Goldberg. Finding a maximum density subgraph. 1984.

Francesco Gullo, Domenico Mandaglio, and Andrea Tagarelli. Neural discovery of balance-aware polarized communities. *Mach. Learn.*, 113(9):6611–6644, 2024.

Francesco Gullo, Domenico Mandaglio, and Andrea Tagarelli. Polarized communities meet densest subgraph: Efficient and effective polarization detection in signed networks. *TKDD*, 20(2), 2026. ISSN 1556-4681. doi: 10.1145/3779064. URL `https://doi.org/10.1145/3779064`.

Frank Harary. On the notion of balance of a signed graph. *Michigan Mathematical Journal*, 2: 143–146, 1953.

Elfarouk Harb, Kent Quanrud, and Chandra Chekuri. Faster and scalable algorithms for densest subgraph and decomposition. *Proc. NIPS Conf.*, 35:26966–26979, 2022.

J. Hastad. Clique is hard to approximate within $n^{1-\epsilon}$. In *Proc. FOCS Symp.*, pp. 627, 1996.

Yixuan He, Gesine Reinert, Songchao Wang, and Mihai Cucuringu. SSSNET: semi-supervised signed network clustering. In *Proc. SDM Conf.*, pp. 244–252, 2022.

Marilena Hohmann, Karel Devriendt, and Michele Coscia. Quantifying ideological polarization on a network using generalized Euclidean distance. *Science Advances*, 9(9), 2023.

Junghoon Kim, Hyun Ji Jeong, Sungsu Lim, and Jungeun Kim. Effective and efficient core computation in signed networks. *Information Sciences*, 634:290–307, 2023.

Jérôme Kunegis. KONECT – The Koblenz Network Collection. In *Proc. WWW Conf.*, pp. 1343–1350, 2013. `http://konect.cc`.

Jérôme Kunegis, Stephan Schmidt, Andreas Lommatzsch, Jürgen Lerner, Ernesto William De Luca, and Sahin Albayrak. Spectral analysis of signed graphs for clustering, prediction and visualization. In *Proc. SDM Conf.*, pp. 559–570, 2010.

Lucio La Cava, Domenico Mandaglio, and Andrea Tagarelli. Polarization in decentralized online social networks. In *Proc. of the ACM Web Science Conference*, pp. 48–52, 2024.

Mirko Lai, Viviana Patti, Giancarlo Ruffo, and Paolo Rosso. Stance evolution and twitter interactions in an Italian political debate. In *Proc. NLDB Conf.*, pp. 15–27, 2018.

Tommaso Lanciano, Atsushi Miyauchi, Adriano Fazzone, and Francesco Bonchi. A survey on the densest subgraph problem and its variants. *ACM Computing Surveys*, 56(8):1–40, 2024.

Victor E. Lee, Ning Ruan, Ruoming Jin, and Charu C. Aggarwal. A survey of algorithms for dense subgraph discovery. In *Managing and Mining Graph Data*, pp. 303–336. 2010.

Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data`, June 2014.

Fragkiskos D Malliaros, Christos Giatsidis, Apostolos N Papadopoulos, and Michalis Vazirgiannis. The core decomposition of networks: Theory, algorithms and applications. *VLDBJ*, 29(1):61–92, 2020.

Pedro Mercado, Francesco Tudisco, and Matthias Hein. Clustering signed networks with the geometric mean of laplacians. In *Proc. NIPS Conf.*, pp. 4421–4429, 2016.

Jason Niu and Ahmet Erdem Sariyüce. On cohesively polarized communities in signed networks. In *Proc. WWW Conf.*, pp. 1339–1347, 2023.

Bruno Ordozgoiti, Antonis Matakos, and Aristides Gionis. Finding large balanced subgraphs in signed networks. In *Proc. WWW Conf.*, pp. 1378–1388, 2020.

James B Orlin. A faster strongly polynomial time algorithm for submodular function minimization. *Mathematical Programming*, 118(2):237–251, 2009.

Seyed-Vahid Sanei-Mehri, Apurba Das, Hooman Hashemi, and Srikanta Tirthapura. Mining largest maximal quasi-cliques. *TKDD*, 15(5):1–21, 2021.

Joao Sedoc, Jean Gallier, Dean Foster, and Lyle Ungar. Semantic word clusters using signed spectral clustering. In *Proc. ACL Conf.*, pp. 939–949, 2017.

Stephen B Seidman. Network structure and minimum degree. *Social networks*, 5(3):269–287, 1983.

Xiaodan Song, Yun Chi, Koji Hino, and Belle Tseng. Identifying opinion leaders in the blogosphere. In *Proc. CIKM Conf.*, pp. 971–974, 2007.

Jiliang Tang, Yi Chang, Charu Aggarwal, and Huan Liu. A survey of signed network mining in social media. 49(3):1–37, 2016.

Charalampos Tsourakakis, Francesco Bonchi, Aristides Gionis, Francesco Gullo, and Maria Tsiarli. Denser than the densest subgraph: extracting optimal quasi-cliques with quality guarantees. In *Proc. KDD Conf.*, pp. 104–112, 2013.

Charalampos E. Tsourakakis. The k-clique densest subgraph problem. In *Proc. WWW Conf.*, pp. 1122–1132, 2015.

Ruo-Chun Tzeng, Bruno Ordozgoiti, and Aristides Gionis. Discovering conflicting groups in signed networks. *Proc. NIPS Conf.*, 33:10974–10985, 2020.

Nate Veldt, Austin R Benson, and Jon Kleinberg. The generalized mean densest subgraph problem. In *Proc. KDD Conf.*, pp. 1604–1614, 2021.

Han Xiao, Bruno Ordozgoiti, and Aristides Gionis. Searching for polarization in signed graphs: a local spectral approach. In *Proc. WWW Conf.*, pp. 362–372, 2020.

Kai Yao, Lijun Chang, and Lu Qin. Computing maximum structural balanced cliques in signed graphs. In *Proc. ICDE Conf.*, pp. 1004–1016, 2022.

Ahmad Zareie and Rizos Sakellariou. Maximizing the diversity of exposure in online social networks by identifying users with increased susceptibility to persuasion. *TKDD*, 18(2):1–21, 2023.

A Preliminaries on Quasi-Arithmetic Means

We provide here the general background on quasi-arithmetic means underlying the exponential mean E_p introduced in Section 3 of the main paper.

Definition 2 (Quasi-arithmetic mean)**.** *Given a continuous, strictly monotone function $\varphi\colon \mathbb{R} \to \mathbb{R}$, the quasi-arithmetic mean (Kolmogorov–Nagumo mean) of a sequence $\mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{R}^n$ is*

$$M_\varphi(\mathbf{a}) \;=\; \varphi^{-1}\!\left(\frac{1}{n}\sum_{i=1}^{n}\varphi(a_i)\right). \tag{7}$$

The values a_i are mapped into a transformed space via φ, averaged there, and mapped back by φ^{-1}. When φ is the identity, M_φ reduces to the arithmetic mean.

Exponential specialization. Setting $\varphi(x) = e^{px}$ for a real parameter $p \neq 0$ yields a strictly monotone function (increasing for $p > 0$, decreasing for $p < 0$) with inverse $\varphi^{-1}(y) = \frac{1}{p}\log y$. Substituting into Eq. 7 gives the exponential mean used throughout this paper:

$$E_p(\mathbf{a}) \;=\; \frac{1}{p}\,\log\left(\frac{1}{n}\sum_{i=1}^{n} e^{p\,a_i}\right), \qquad p \neq 0, \tag{8}$$

Unlike the standard power mean $M_p(\mathbf{a}) = \left(\frac{1}{n}\sum_i a_i^p\right)^{1/p}$, which requires $a_i > 0$ and for which even exponents suppress sign information, E_p is well-defined for arbitrary $\mathbf{a} \in \mathbb{R}^n$.

Limiting behavior. E_p recovers three fundamental summary statistics in the limiting cases of p.

Case $p \to 0$ (arithmetic mean). Expanding $e^{p\,a_i} = 1 + p\,a_i + O(p^2)$ and $\log(1+x) = x + O(x^2)$:

$$E_p(\mathbf{a}) = \frac{1}{p}\,\log\left(1 + \frac{p}{n}\sum_i a_i + O(p^2)\right) = \frac{1}{n}\sum_{i=1}^{n} a_i + O(p).$$

Hence $E_0(\mathbf{a}) = \lim_{p\to 0} E_p(\mathbf{a}) = \frac{1}{n}\sum_{i=1}^{n} a_i$.

Case $p \to +\infty$ (maximum). Let $a_{\max} = \max_i a_i$ and let k be the number of indices attaining it. Factoring $e^{p\,a_{\max}}$ out of the sum:

$$E_p(\mathbf{a}) = a_{\max} + \frac{1}{p}\,\log\left(\frac{1}{n}\sum_{i=1}^{n} e^{p(a_i - a_{\max})}\right).$$

Every term with $a_i < a_{\max}$ vanishes as $p \to +\infty$, so the argument of the logarithm tends to k/n and the correction $\frac{1}{p}\log(k/n) \to 0$. Thus $E_{+\infty}(\mathbf{a}) = \lim_{p\to+\infty} E_p(\mathbf{a}) = \max_i a_i$.

Case $p \to -\infty$ (minimum). Let $a_{\min} = \min_i a_i$. Factoring $e^{p\,a_{\min}}$ and noting that $p(a_i - a_{\min}) \to -\infty$ for every $a_i > a_{\min}$ when $p \to -\infty$, the same reasoning gives $E_{-\infty}(\mathbf{a}) = \lim_{p\to-\infty} E_p(\mathbf{a}) = \min_i a_i$.

E_p continuously interpolates from the minimum ($p \to -\infty$) through the arithmetic mean ($p = 0$) to the maximum ($p \to +\infty$), while remaining well-defined for sequences containing negative values. This is the property exploited in Definition 1 to define the p-polarity objective for signed graphs.

Example 2. For $\mathbf{a} = (5, -6, -4, 2)$: $E_0(\mathbf{a}) = -0.75$, $E_1(\mathbf{a}) \approx 3.66$, $E_2(\mathbf{a}) \approx 4.3$, $E_{+\infty}(\mathbf{a}) = 5$ (the maximum). For negative p: $E_{-1}(\mathbf{a}) \approx -4.74$, $E_{-2}(\mathbf{a}) \approx -5.31$, $E_{-\infty}(\mathbf{a}) = -6$ (the minimum). □

B Proofs

Theorem 2 Max-2pc can be solved in polynomial time.

Proof. Let $w = \arg\max_{u\in V}\big(d^+(u) + d^-(u)\big)$ and define $S^w_{1,2} = \{S^w_1, S^w_2\}$ with $S^w_1 = \{w\} \cup N^+(w)$ and $S^w_2 = N^-(w)$. Since G has no multi-edges, $N^+(w) \cap N^-(w) = \emptyset$ and $w \notin N^-(w)$, so $S^w_1 \cap S^w_2 = \emptyset$ and $S^w_{1,2}$ is feasible.

Key observation. For any feasible solution $S'_{1,2} = \{S'_1, S'_2\}$ and any node $u \in S'_1 \cup S'_2$:

$$d^{\pm}_{S'_{1,2}}(u) = d^{+}_{S'_{1,2}}(u) - d^{-}_{S'_{1,2}}(u) \le d^{+}_{S'_{1,2}}(u) \le d^{+}(u) + d^{-}(u),$$

where the first inequality holds because $d^{-}_{S'_{1,2}}(u) \ge 0$, and the second because the consistent degree counts only edges whose both endpoints lie in $S'_1 \cup S'_2$, hence it cannot exceed $d^{+}(u) + d^{-}(u)$.

In $S^w_{1,2}$, every positive neighbor of w belongs to S^w_1 (same community) and every negative neighbor to S^w_2 (opposite community), so all incident edges of w are consistent:

$$d^{\pm}_{S^w_{1,2}}(w) = \big(d^{+}(w) + d^{-}(w)\big) - 0 = d^{+}(w) + d^{-}(w).$$

Optimality. For every feasible $S'_{1,2}$ and every $u \in S'_1 \cup S'_2$, chaining the key observation with the definition of w gives:

$$d^{\pm}_{S'_{1,2}}(u) \le d^{+}(u) + d^{-}(u) \le d^{+}(w) + d^{-}(w) = d^{\pm}_{S^w_{1,2}}(w).$$

Taking the maximum over $u \in S'_1 \cup S'_2$ yields $\max_{u \in S'_1 \cup S'_2} d^{\pm}_{S'_{1,2}}(u) \le d^{\pm}_{S^w_{1,2}}(w)$, so $S^w_{1,2}$ is optimal. Moreover, both the computation of w and the construction of $S^w_{1,2}$ can be performed in polynomial time, thus the MAX-2PC problem admits a polynomial-time solution. □

Theorem 3 CORE-2PC is **NP**-hard.

We prove Theorem 3 using the following construction and two intermediate lemmas.

Decision problems. Given a signed graph and an integer threshold τ, the decision version of CORE-2PC asks whether there exist disjoint communities $S_{1,2} = \{S_1, S_2\}$, with $S_1 \cup S_2 \neq \emptyset$, such that $d^{\pm}_{S_{1,2}}(u) \ge \tau$ for every $u \in S_1 \cup S_2$. We call these vertices *selected*; all other vertices are neutral, and edges incident to them are ignored. Throughout this proof, a solution is *feasible* if it satisfies this threshold condition.

We reduce from *Satisfactory Partition*, which is NP-complete (Gerber & Kobler, 2000). Given a simple undirected graph $H = (V_H, E_H)$, this problem asks for a partition (V_1, V_2) of V_H into two nonempty parts such that

$$d_{V_a}(v) \ge d_{V_{3-a}}(v) \qquad \text{for every } v \in V_a, \quad a \in \{1, 2\},$$

where $d_X(v)$ is the number of neighbors of v in X.

Construction. Write $V_H = \{v_1, \dots, v_n\}$ and let Δ_H be the maximum degree of H. We may assume $n \ge 3$, since excluding the finitely many smaller instances does not affect NP-hardness. Set

$$q = n, \qquad \tau = 6q, \qquad \text{so that} \qquad 3q + \Delta_H < \tau. \tag{9}$$

For two distinct vertices x, y, define an *agreement gadget* $R^{=}(x, y)$ and a *disagreement gadget* $R^{\neq}(x, y)$ attached to them. Each has $\tau - 2$ new vertices forming a positive clique and three new, pairwise nonadjacent bridge vertices. Each bridge is joined positively to every clique vertex and to x. Its edge to y is positive in $R^{=}$ and negative in $R^{\neq}$. There are no other edges incident to these internal vertices. Consequently, each internal vertex has unsigned degree exactly τ:

$$(\tau - 3) + 3 = \tau \quad \text{for a clique vertex,} \qquad (\tau - 2) + 2 = \tau \quad \text{for a bridge vertex.}$$

For each $j \in \{1, \dots, n\}$, construct a separate graph G_j. It contains an *original vertex* v'_k for every $v_k \in V_H$ and a positive edge for every edge of H. Consider the cyclic pairs $\Pi_k = (v'_k, v'_{k+1})$, with indices modulo n; these pairs only index gadget attachments and do not add edges between original vertices. For every $k \neq j$, attach q copies of each of $R^{=}(v'_k, v'_{k+1})$ and $R^{\neq}(v'_k, v'_{k+1})$. On the *designated pair* Π_j, attach only q copies of $R^{\neq}(v'_j, v'_{j+1})$. All gadget copies have disjoint internal vertex sets. Each G_j has its own original vertices; we suppress the component index in their notation.

The output is the disjoint union $G = \cup_{j=1}^{n} G_j$, with threshold τ. Each gadget has $O(n)$ vertices and $O(n^2)$ edges, and each G_j contains $O(n^2)$ gadgets. Thus G has $O(n^4)$ vertices and $O(n^5)$ edges and is constructible in polynomial time.

For a gadget copy R, let $W(R)$ denote its internal vertex set. For a pair of communities $S_{1,2} = \{S_1, S_2\}$, write $U = S_1 \cup S_2$. We call R *selected* if $W(R) \cap U \neq \emptyset$. A gadget attached to selected vertices x, y is *compatible* with their assignments if it is an agreement gadget and x, y share a community, or a disagreement gadget and x, y lie in opposite communities.

Lemma 2. *Let $S_{1,2}$ be a feasible solution and let R be a gadget copy attached to the ordered pair (x, y). Then either $W(R) \cap U = \emptyset$, or there exists $i \in \{1, 2\}$ such that*

$$W(R) \cup \{x\} \subseteq S_i, \qquad \begin{cases} y \in S_i, & \text{if } R \text{ is an agreement gadget,} \\ y \in S_{3-i}, & \text{if } R \text{ is a disagreement gadget.} \end{cases} \tag{10}$$

Conversely, for any community assignment satisfying Eq. 10, every vertex in $W(R)$ has net degree balance exactly τ, and the edges of R contribute $+3$ to the net degree balance of each of x and y.

Proof. Suppose $W(R) \cap U \neq \emptyset$ and choose $w \in W(R) \cap U$. Since $S_{1,2}$ is feasible and $\deg(w) = \tau$,

$$\tau \leq d^{\pm}_{S_{1,2}}(w) \leq d^{+}_{S_{1,2}}(w) \leq \deg(w) = \tau.$$

Equality forces all its neighbors to be selected and all incident edges to be consistent. Applying the same argument throughout the connected graph induced by $W(R)$ shows that $W(R) \subseteq U$ and every edge incident to an internal vertex is consistent. In particular, $x, y \in U$.

All edges within $W(R)$ are positive, so $W(R) \subseteq S_i$ for some $i \in \{1, 2\}$. The positive bridge edges to x give $x \in S_i$. The bridge edges to y give $y \in S_i$ for an agreement gadget and $y \in S_{3-i}$ for a disagreement gadget, establishing Eq. 10.

Conversely, under any assignment satisfying Eq. 10, all edges of R are consistent and each internal vertex has all its τ neighbors selected. Thus its net degree balance is τ. Each of x and y has exactly three incident edges belonging to R, all consistent, so their contributions are $+3$ each. □

For a fixed component G_j, let $O_j = \{v'_1, \ldots, v'_n\}$. For each cyclic pair Π_k, let r_k be the number of gadget copies R attached to Π_k with $W(R) \cap U \neq \emptyset$.

Lemma 3. *Let $S_{1,2}$ be a feasible solution. If $U \cap V(G_j) \neq \emptyset$, then:*

1. *$O_j \subseteq U$;*
2. *$|\{v'_j, v'_{j+1}\} \cap S_i| = 1$ for each $i \in \{1, 2\}$;*
3. *$1 \leq r_k \leq q$ for every $k \in \{1, \ldots, n\}$.*

Consequently, the total contribution g_k of edges from v'_k to internal gadget vertices satisfies

$$g_k = 3(r_{k-1} + r_k) \leq 6q = \tau,$$

where indices are taken modulo n.

Proof. First, $r_k \leq q$ for every k. Indeed, if either vertex of Π_k is neutral, Lemma 2 implies $r_k = 0$. Otherwise, the same lemma implies that only compatible gadgets can be selected. At most one gadget type is compatible with the two community assignments, and the construction contains at most q copies of that type on Π_k.

Now consider any selected original vertex v'_k. Let b_k be the contribution from its edges to selected original vertices. By Lemma 2, each selected gadget attached to v'_k contributes $+3$, so

$$d^{\pm}_{S_{1,2}}(v'_k) = 3(r_{k-1} + r_k) + b_k, \qquad b_k \leq \Delta_H. \tag{11}$$

If $r_{k-1} = 0$ or $r_k = 0$, the upper bound on the other term and Eqs. 9 and 11 give

$$d^{\pm}_{S_{1,2}}(v'_k) \leq 3q + \Delta_H < \tau,$$

contradicting feasibility. Therefore $r_{k-1}, r_k \geq 1$, and Lemma 2 implies that both v'_{k-1} and v'_{k+1} are selected.

Since $U \cap V(G_j) \neq \emptyset$, either an original vertex is selected or a selected internal vertex forces its two attached original vertices to be selected by Lemma 2. The preceding argument then propagates

selection around the cyclic order, proving $O_j \subseteq U$. Applying it to every original vertex also gives $r_k \geq 1$ for every k.

In particular, $r_j \geq 1$. Only disagreement gadgets are attached to Π_j, so Lemma 2 places v'_j and v'_{j+1} in opposite communities. Finally, the same lemma gives $g_k = 3(r_{k-1} + r_k)$, which is at most $6q = \tau$ because $r_{k-1}, r_k \leq q$. □

Theorem 3. CORE-2PC is NP-hard.

Proof. We show that H admits a satisfactory partition if and only if (G, τ) is a YES-instance of the decision version of CORE-2PC.

($\Rightarrow$) Let (V_1, V_2) be a satisfactory partition of H. Since both parts are nonempty, some consecutive pair v_j, v_{j+1} in the cyclic order lies in opposite parts. In G_j, assign each original vertex to the community corresponding to its part. On every cyclic pair, select all q compatible gadget copies and place their internal vertices in the community of v'_k for gadgets attached to $\Pi_k = (v'_k, v'_{k+1})$. These copies exist on the designated pair as well, since v'_j and v'_{j+1} lie in opposite communities. Leave every other vertex of G neutral.

By the converse assertion of Lemma 2, each selected internal vertex has balance exactly τ. Each original vertex receives contribution $3q$ from each of its two incident pairs. Thus, for $v_k \in V_a$,

$$d^{\pm}_{S_{1,2}}(v'_k) = 6q + d_{V_a}(v_k) - d_{V_{3-a}}(v_k) \geq 6q = \tau.$$

The constructed solution is nonempty and feasible.

($\Leftarrow$) Let $S_{1,2}$ be a nonempty feasible solution. Choose a graph G_j containing a selected vertex and discard all selected vertices outside G_j. Since the union is disjoint, this preserves all remaining selected vertices' balances.

By Lemma 3, all original vertices of G_j are selected and the designated pair has endpoints in opposite communities. Therefore

$$V_a = \{v_k : v'_k \in S_a\}, \qquad a \in \{1, 2\},$$

defines a partition of V_H into two nonempty parts.

For $v_k \in V_a$, the positive edges inherited from H contribute $d_{V_a}(v_k) - d_{V_{3-a}}(v_k)$. The gadget contribution g_k is at most τ by Lemma 3, so

$$\begin{aligned} \tau &\leq d^{\pm}_{S_{1,2}}(v'_k) \\ &= g_k + d_{V_a}(v_k) - d_{V_{3-a}}(v_k) \\ &\leq \tau + d_{V_a}(v_k) - d_{V_{3-a}}(v_k). \end{aligned}$$

Hence $d_{V_a}(v_k) \geq d_{V_{3-a}}(v_k)$ for every vertex, proving that (V_1, V_2) is a satisfactory partition.

The construction is therefore a polynomial-time many-one reduction. Moreover, the target decision problem belongs to NP: a certificate assigns each vertex to one of the two communities or the neutral set, and its balances can be checked in linear time. Thus the decision version of CORE-2PC is NP-complete, and the optimization problem is NP-hard. □

Proposition 1 Let $G = (V, E^+, E^-)$ be a balanced signed graph. Then, for every optimal solution $\{S_1^*, S_2^*\}$ for the P-MEAN-2PC problem there exists a bipartition (V_1, V_2) of V such that $S_1^* \subseteq V_1$ and $S_2^* \subseteq V_2$.

Proof. Since G is balanced, a bipartition (V_1, V_2) of V exists and can be computed in linear time.

Given any feasible solution $S_{1,2} = \{S_1, S_2\}$, let $U = S_1 \cup S_2$ and define $S^*_{1,2} = \{S_1^*, S_2^*\} = \{U \cap V_1,\ U \cap V_2\}$. Since (V_1, V_2) partitions V, we have $S_1^* \cap S_2^* = \emptyset$, $S_1^* \cup S_2^* = U$, and $S_1^* \subseteq V_1$, $S_2^* \subseteq V_2$, so $S^*_{1,2}$ is a valid feasible solution with the same selected node set U.

We claim that for every $u \in U$:

$$d^{\pm}_{S^*_{1,2}}(u) \ \geq \ d^{\pm}_{S_{1,2}}(u), \tag{12}$$

with strict inequality if and only if u is incident to at least one inconsistent edge in $S_{1,2}$.

Consider any edge (u, v) with $u, v \in U$. We show it is consistent with $S^*_{1,2}$:

- If $(u, v) \in E^+$: the balance property of G implies u and v lie in the same part of (V_1, V_2), so they belong to the same community in $S^*_{1,2}$. A positive intra-community edge is consistent.
- If $(u, v) \in E^-$: the balance property implies u and v lie in different parts, so they belong to different communities in $S^*_{1,2}$. A negative inter-community edge is consistent.

Hence every edge incident to u within U is consistent with $S^*_{1,2}$, giving $d^-_{S^*_{1,2}}(u) = 0$, so $d^\pm_{S^*_{1,2}}(u) = d_U(u)$, where $d_U(u)$ denotes the induced degree of u in the unsigned graph $G[U]$. Each inconsistent edge of u in $S_{1,2}$ contributes -1 rather than $+1$ to $d^\pm_{S_{1,2}}(u)$, so

$$d^\pm_{S_{1,2}}(u) \;=\; d_U(u) - 2\, d^-_{S_{1,2}}(u) \;\le\; d_U(u) \;=\; d^\pm_{S^*_{1,2}}(u),$$

establishing Eq. 12. If $S_{1,2}$ contains an inconsistent edge (u_0, v_0), then $d^\pm_{S_{1,2}}(u_0) < d^\pm_{S^*_{1,2}}(u_0)$ and $d^\pm_{S_{1,2}}(v_0) < d^\pm_{S^*_{1,2}}(v_0)$ strictly.

Since $S_{1,2}$ and $S^*_{1,2}$ share the same node set U, only the means of the per-node balance sequences differ. We show $E_p(S_{1,2}) \le E_p(S^*_{1,2})$ for all p, with strict inequality whenever $S_{1,2}$ has at least one inconsistent edge.

- *Case $p > 0$:* Since e^{px} is strictly increasing, Eq. 12 gives $e^{p\, d^\pm_{S_{1,2}}(u)} \le e^{p\, d^\pm_{S^*_{1,2}}(u)}$ for all $u \in U$, with strict inequality for u_0. Summing over U and dividing by $|U|$ yields a strict inequality between the two exponential means. Since $(1/p)\log(\cdot)$ is strictly increasing, we conclude $E_p(S_{1,2}) < E_p(S^*_{1,2})$.
- *Case $p < 0$:* Since e^{px} is strictly decreasing, Eq. 12 gives $e^{p\, d^\pm_{S_{1,2}}(u)} \ge e^{p\, d^\pm_{S^*_{1,2}}(u)}$ for all $u \in U$, with strict inequality for u_0. Thus the mean of the exponentials is strictly larger for $S_{1,2}$ than for $S^*_{1,2}$. Since $(1/p)\log(\cdot)$ is strictly decreasing, we conclude $E_p(S_{1,2}) < E_p(S^*_{1,2})$.
- *Case $p = 0$:* E_0 is the arithmetic mean and Eq. 12 gives directly $E_0(S_{1,2}) < E_0(S^*_{1,2})$.
- *Cases $p \to \pm\infty$:* Inequality 12 implies

  $$\max_{u\in U} d^\pm_{S^*_{1,2}}(u) \;\ge\; \max_{u\in U} d^\pm_{S_{1,2}}(u), \qquad \min_{u\in U} d^\pm_{S^*_{1,2}}(u) \;\ge\; \min_{u\in U} d^\pm_{S_{1,2}}(u),$$

  so $E_{\pm\infty}(S_{1,2}) \le E_{\pm\infty}(S^*_{1,2})$.

Concluding, if an optimal solution $S_{1,2}$ contained any inconsistent edge, the canonical solution $S^*_{1,2}$ would achieve a strictly higher p-polarity, contradicting optimality. Therefore every optimal solution is free of inconsistent edges: every positive edge between nodes of U is intra-community and every negative edge between nodes of U is inter-community. This is precisely the condition that S_1 and S_2 each lie within a single part of the bipartition restricted to U. Since the orientation of the bipartition per connected component of $G[U]$ can be chosen freely, there exists a bipartition (V_1, V_2) of V such that $S_1 \subseteq V_1$ and $S_2 \subseteq V_2$. □

Theorem 4 Let $G = (V, E^+, E^-)$ be a balanced signed graph. For $p \in \{-\infty, 0\}$, P-MEAN-2PC can be solved in polynomial time.

Proof. If the input graph $G = (V, E^+, E^-)$ is balanced, there exists a balanced bipartition (V_1, V_2). Given a bipartition (V_1, V_2), for any $S \subseteq V$, we denote by $\widetilde{S}_{1,2} := \{S \cap V_1,\ S \cap V_2\}$ the *canonical* solution obtained by projecting S onto the bipartition. Equivalently, letting $\widetilde{S}_1 := S \cap V_1$ and $\widetilde{S}_2 := S \cap V_2$, we have $\widetilde{S}_{1,2} = \{\widetilde{S}_1, \widetilde{S}_2\}$.

Proposition 1 establishes that optimal solutions on balanced graphs may be assumed consistent with a balanced bipartition. Here we need the stronger statement that this can be done with respect to the fixed balanced bipartition (V_1, V_2) used for the reduction. We prove this next.

Consider any feasible solution $S_{1,2} = \{S_1, S_2\}$ to P-MEAN-2PC with overall node set $S = S_1 \cup S_2 \neq \emptyset$. Let $\widetilde{S}_{1,2} = \{\widetilde{S}_1, \widetilde{S}_2\} = \{S \cap V_1,\ S \cap V_2\}$ be the canonical solution obtained by projecting S onto (V_1, V_2). We first show that this projection to (V_1, V_2) does not worsen the objective when

$p \in \{-\infty, 0\}$. Since G is balanced, every edge with both endpoints in S is consistent with (V_1, V_2), hence it is consistent with $\widetilde{S}_{1,2}$. For any $u \in S$, each incident edge (u, v), with $v \in S$, therefore contributes $+1$ to $d^{\pm}_{\widetilde{S}_{1,2}}(u)$ while it contributes either $+1$ (if consistent) or -1 (if inconsistent) to $d^{\pm}_{S_{1,2}}(u)$. Summing over all incident edges yields $d^{\pm}_{\widetilde{S}_{1,2}}(u) \geq d^{\pm}_{S_{1,2}}(u)$ for all $u \in S$.

As $\widetilde{S}_1 \cup \widetilde{S}_2 = S$, the objective (which for $p = 0$ is the average of $d^{\pm}_{S_{1,2}}(u)$ over S, and for $p = -\infty$ is its minimum) cannot decrease under this reassignment. Hence an optimal solution exists that is consistent with (V_1, V_2), i.e., of the form $\{U \cap V_1, U \cap V_2\}$, for some nonempty $U \subseteq V$.

Now fix any nonempty $U \subseteq V$ and consider the pair of polarized communities $\widetilde{U}_{1,2} = \{U \cap V_1, U \cap V_2\}$. By the balance property, all edges induced by U are consistent, thus for every $u \in U$, we have $d^{\pm}_{\widetilde{U}_{1,2}}(u) = d_U(u)$, where $d_U(u)$ is the (unsigned) induced degree in the graph $G' = (V, E^+ \cup E^-)$. Therefore, for $p = 0$, $E_0(\widetilde{U}_{1,2}) = 1/|U| \sum_{u \in U} d_U(u)$, i.e. the P-MEAN-2PC objective coincides with the densest-subgraph objective on the unsigned graph, and for $p = -\infty$, $E_{-\infty}(\widetilde{U}_{1,2}) = \min_{u \in U} d_U(u)$, i.e. the MAXCORE objective.

Let U^* be the vertex set returned by DS (when $p = 0$) or by MAXCORE (when $p = -\infty$) on the unsigned graph $G' = (V, E^+ \cup E^-)$. By the equivalences above and the fact that an optimum can be assumed consistent, the lifted pair $\widetilde{U}^*_{1,2} = \{U^* \cap V_1, U^* \cap V_2\}$ is optimal for P-MEAN-2PC. Since both DS and MAXCORE are solvable in polynomial time and the lifting step is linear, P-MEAN-2PC is polynomial-time solvable on balanced inputs for $p \in \{-\infty, 0\}$. □

Proposition 2 Consider the P-MEAN-2PC problem with $p = 0$. Let $S_{1,2} = \{S_1, S_2\}$ be a feasible solution such that $E_0(S_{1,2}) > 0$. If there exists a node $z \in S_1 \cup S_2$ with $d^{\pm}_{S_{1,2}}(z) \leq 0$, then removing z from its community in $S_{1,2}$ yields a new feasible solution $S'_{1,2}$ such that $E_0(S'_{1,2}) > E_0(S_{1,2})$.

Proof. Let us assume w.l.o.g. that $z \in S_1$ and let $S'_{1,2} = \{S_1 \setminus z, S_2\}$ be the P-MEAN-2PC solution obtained by removing z from its community S_1. For $p = 0$, the objective function defined in Eq. (4) simplifies to the arithmetic mean of the net degree balances, i.e., $E_0(S_{1,2}) = \frac{1}{|S_1 \cup S_2|} \sum_{u \in S_1 \cup S_2} d^{\pm}_{S_{1,2}}(u)$.

Let $\mathcal{D}(S_{1,2}) = \sum_{u \in S_1 \cup S_2} d^{\pm}_{S_{1,2}}(u)$ denote the total net degree balance values. Consider the removal of node z. The new total net degree balance score $\mathcal{D}(S'_{1,2})$ differs from $\mathcal{D}(S_{1,2})$ due to two factors: (i) the removal of the term $d^{\pm}_{S_{1,2}}(z)$ from the numerator's summation, and (ii) the update of the net degree balances for all neighbors of z. Let $N_{S_{1,2}}(z) \subset S_1 \cup S_2$ denote the set of neighbors of z in the solution $S_{1,2}$, i.e. $N_{S_{1,2}}(z) = N(z) \cap (S_1 \cup S_2)$. Clearly, it holds that $N_{S_{1,2}}(z) = N^+_{S_{1,2}}(z) \cup N^-_{S_{1,2}}(z)$.

Upon removing z, each neighbor $v \in N^+_{S_{1,2}}(z)$ (resp., $v \in N^-_{S_{1,2}}(z)$) loses one consistent (resp., inconsistent) incident edge, hence its balance decreases (resp., increases) by one, i.e., $d^{\pm}_{S'_{1,2}}(v) = d^{\pm}_{S_{1,2}}(v) - 1$ for $v \in N^+_{S_{1,2}}(z)$ and $d^{\pm}_{S'_{1,2}}(v) = d^{\pm}_{S_{1,2}}(v) + 1$ for $v \in N^-_{S_{1,2}}(z)$.

The total change contributed by the neighbors is the sum of these updates:

$$\begin{aligned}
\sum_{v \in N_{S_{1,2}}(z)} (d^{\pm}_{S'_{1,2}}(v) - d^{\pm}_{S_{1,2}}(v)) &= \sum_{v \in N^+_{S_{1,2}}(z)} (-1) + \sum_{v \in N^-_{S_{1,2}}(z)} (+1) \\
&= -(|N^+_{S_{1,2}}(z)| - |N^-_{S_{1,2}}(z)|) \\
&= -(d^+_{S_{1,2}}(z) - d^-_{S_{1,2}}(z)) \\
&= -d^{\pm}_{S_{1,2}}(z).
\end{aligned}$$

Thus, the total change in the numerator of the objective is $\Delta\mathcal{D} = \mathcal{D}(S'_{1,2}) - \mathcal{D}(S_{1,2}) = -d^{\pm}_{S_{1,2}}(z) - d^{\pm}_{S_{1,2}}(z) = -2d^{\pm}_{S_{1,2}}(z)$. The new objective value is:

$$E_0(S'_{1,2}) = \frac{\mathcal{D}(S_{1,2}) - 2d^{\pm}_{S_{1,2}}(z)}{|S_1| - 1 + |S_2|}.$$

By hypothesis, $d^{\pm}_{S_{1,2}}(z) \leq 0$, which implies $-2d^{\pm}_{S_{1,2}}(z) \geq 0$. Since $E_0(S_{1,2}) > 0$, we have $\mathcal{D}(S_{1,2}) > 0$. We are therefore comparing the original fraction $\frac{\mathcal{D}(S_{1,2})}{|S_1|+|S_2|}$ with a new fraction where the numerator has not decreased and the denominator has strictly decreased (and remains positive, assuming $|S_1| + |S_2| > 1$). It follows immediately that:

$$E_0(S'_{1,2}) = \frac{\mathcal{D}(S_{1,2}) - 2d^{\pm}_{S_{1,2}}(z)}{|S_1| - 1 + |S_2|} \geq \frac{\mathcal{D}(S_{1,2})}{|S_1| - 1 + |S_2|} > \frac{\mathcal{D}(S_{1,2})}{|S_1| + |S_2|} = E_0(S_{1,2}).$$

□

Proposition 3 As $p \to 0^+$, it holds that $\arg\min_{z \in S_1 \cup S_2} \Delta_p(z; S_{1,2}) = \arg\min_{z \in S_1 \cup S_2} d^{\pm}_{S_{1,2}}(z)$ and, as $p \to 0^-$, it holds that $\arg\max_{z \in S_1 \cup S_2} \Delta_p(z; S_{1,2}) = \arg\min_{z \in S_1 \cup S_2} d^{\pm}_{S_{1,2}}(z)$.

Proof. We expand each term of Eq. 6 using the first-order Taylor expansion $e^{px} = 1 + px + O(p^2)$ as $p \to 0$.

The first term gives:

$$e^{p \cdot d^{\pm}_{S_{1,2}}(z)} = 1 + p \cdot d^{\pm}_{S_{1,2}}(z) + O(p^2).$$

For each $u \in N^+_{S_{1,2}}(z)$, the second term contributes:

$$e^{p \cdot d^{\pm}_{S_{1,2}}(u)} - e^{p(d^{\pm}_{S_{1,2}}(u)-1)} = \Big(1 + p \cdot d^{\pm}_{S_{1,2}}(u)\Big) - \Big(1 + p(d^{\pm}_{S_{1,2}}(u) - 1)\Big) + O(p^2) = p + O(p^2),$$

so summing over all $u \in N^+_{S_{1,2}}(z)$ yields $p \cdot d^+_{S_{1,2}}(z) + O(p^2)$. Similarly, for each $u \in N^-_{S_{1,2}}(z)$, the third term contributes:

$$e^{p \cdot d^{\pm}_{S_{1,2}}(u)} - e^{p(d^{\pm}_{S_{1,2}}(u)+1)} = \Big(1 + p \cdot d^{\pm}_{S_{1,2}}(u)\Big) - \Big(1 + p(d^{\pm}_{S_{1,2}}(u) + 1)\Big) + O(p^2) = -p + O(p^2),$$

so summing over all $u \in N^-_{S_{1,2}}(z)$ yields $-p \cdot d^-_{S_{1,2}}(z) + O(p^2)$.

Combining all three contributions:

$$\Delta_p(z; S_{1,2}) = 1 + p \cdot d^{\pm}_{S_{1,2}}(z) + p \cdot d^+_{S_{1,2}}(z) - p \cdot d^-_{S_{1,2}}(z) + O(p^2).$$

Since $d^+_{S_{1,2}}(z) - d^-_{S_{1,2}}(z) = d^{\pm}_{S_{1,2}}(z)$, this simplifies to:

$$\Delta_p(z; S_{1,2}) = 1 + 2p \cdot d^{\pm}_{S_{1,2}}(z) + O(p^2).$$

Since the constant term 1 is the same for all nodes z, the ranking induced by $\Delta_p(z; S_{1,2})$ over nodes in $S_{1,2}$ is determined, as $p \to 0$, entirely by the coefficient $2p \cdot d^{\pm}_{S_{1,2}}(z)$. Recall that GenPeel-$p$-2PC selects the node z which minimizes (resp. maximizes) $\Delta_p(z; S_{1,2})$ for $p > 0$ (resp. $p < 0$). Therefore:

$$\lim_{p \to 0^+} \arg\min_{z \in S_1 \cup S_2} \Delta_p(z; S_{1,2}) = \arg\min_{z \in S_1 \cup S_2} d^{\pm}_{S_{1,2}}(z),$$
$$\lim_{p \to 0^-} \arg\max_{z \in S_1 \cup S_2} \Delta_p(z; S_{1,2}) = \arg\min_{z \in S_1 \cup S_2} d^{\pm}_{S_{1,2}}(z),$$

and in both cases GenPeel-p-2PC selects the node with the lowest net degree balance, which completes the proof. □

Lemma 4. *Let $S_{1,2} = \{S_1, S_2\}$ be a nonempty solution. Then, for every finite $p > 0$, it holds that*

$$E_{+\infty}(S_{1,2}) - \frac{\log |S_1 \cup S_2|}{p} \;\leq\; E_p(S_{1,2}) \;\leq\; E_{+\infty}(S_{1,2}).$$

Proof. Write $a_u = d^{\pm}_{S_{1,2}}(u)$ for $u \in S_1 \cup S_2$ and $a_{\max} = \max_u a_u$, so that $E_{+\infty}(S_{1,2}) = a_{\max}$. Since $e^{p \cdot a_u} \leq e^{p \cdot a_{\max}}$ for every u, we get $\frac{1}{|S_1 \cup S_2|} \sum_u e^{p \cdot a_u} \leq e^{p \cdot a_{\max}}$ and, as $\frac{1}{p}\log(\cdot)$ is increasing for $p > 0$, $E_p(S_{1,2}) \leq a_{\max}$. Conversely, keeping only the maximizing term, $\frac{1}{|S_1 \cup S_2|} \sum_u e^{p \cdot a_u} \geq \frac{1}{|S_1 \cup S_2|} e^{p \cdot a_{\max}}$, whence $E_p(S_{1,2}) \geq a_{\max} - \frac{\log |S_1 \cup S_2|}{p}$. □

Proposition 5. *Let $G = (V, E^+, E^-)$ be a signed graph, let $p > 0$ be finite, and let $S^*_{1,2}$ be an optimal solution of* P-MEAN-2PC *for that value of p. Then the solution $\widehat{S}_{1,2}$ returned by* Max-2PC-Exact *satisfies $E_p(\widehat{S}_{1,2}) \geq E_p(S^*_{1,2}) - \frac{\log |V|}{p}$.*

Proof. By Theorem 2, Max-2PC-Exact returns a solution $\widehat{S}_{1,2}$ that is optimal for MAX-2PC, i.e., $E_{+\infty}(\widehat{S}_{1,2}) \geq E_{+\infty}(S_{1,2})$ for every feasible $S_{1,2}$. Using Lemma 4 twice, and $|S_1 \cup S_2| \leq |V|$ for every solution $S_{1,2}$,

$$E_p(\widehat{S}_{1,2}) \;\geq\; E_{+\infty}(\widehat{S}_{1,2}) - \frac{\log |V|}{p} \;\geq\; E_{+\infty}(S^*_{1,2}) - \frac{\log |V|}{p} \;\geq\; E_p(S^*_{1,2}) - \frac{\log |V|}{p},$$

where the second inequality uses the optimality of $\widehat{S}_{1,2}$ for $E_{+\infty}$, and the third the upper bound $E_p(S_{1,2}) \leq E_{+\infty}(S_{1,2})$ of Lemma 4. □

C ADDITIONAL DISCUSSION ON THE BALANCED GRAPH CASE.

Next, we complement the discussion in the main paper on the polynomial-time solvability of P-MEAN-2PC for $p > 0$ when the input signed graph is balanced.

Recall that, for any finite $p > 0$, maximizing $E_p(S_{1,2})$ is equivalent to maximizing $e^{p \cdot E_p(S_{1,2})}$. In other words, the P-MEAN-2PC problem seeks the pair of polarized communities with the highest average pth-exponential-power net degree balance:

$$f_p(S_{1,2}) = \frac{1}{|S_1 \cup S_2|} \sum_{u \in S_1 \cup S_2} e^{p \cdot d^{\pm}_{S_{1,2}}(u)} \tag{13}$$

For this reason, we focus on the latter next. Given $\alpha > 0$, the corresponding decision problem asks whether there exists a feasible solution $S_{1,2} = \{S_1, S_2\}$ such that $f_p(S_{1,2}) \geq \alpha$, or equivalently,

$$\psi_{p,\alpha}(S_{1,2}) \;:=\; \sum_{u \in S_1 \cup S_2} e^{p \cdot d^{\pm}_{S_{1,2}}(u)} \;-\; \alpha |S_1 \cup S_2| \;\geq\; 0.$$

We can answer this question by solving $\max_{S_{1,2}=\{S_1,S_2\}} \psi_{p,\alpha}(S_{1,2})$.

Since G is balanced, fix a bipartition (V_1, V_2). Since on a balanced graph the canonical solution makes all edges consistent, it dominates any non-canonical assignment for any fixed U (as e^{px} is increasing for $p > 0$); we may therefore restrict to solutions consistent with (V_1, V_2). Such balanced solutions are uniquely determined by their selected set $U := S_1 \cup S_2$ via $S_{1,2} = \{U \cap V_1,\ U \cap V_2\}$. Therefore,

$$\max_{S_{1,2}=\{S_1,S_2\}} \psi_{p,\alpha}(S_{1,2}) \;=\; \max_{\emptyset \neq U \subseteq V} \psi_{p,\alpha}(\{U \cap V_1,\ U \cap V_2\}) \;=\; \max_{\emptyset \neq U \subseteq V} \Phi_{p,\alpha}(U),$$

where the induced set function $\Phi_{p,\alpha} : 2^V \to \mathbb{R}$ is defined as

$$\Phi_{p,\alpha}(U) \;:=\; \sum_{u \in U} e^{p \cdot d_U(u)} \;-\; \alpha |U|,$$

and $d_U(u)$ denotes the degree of u in the induced subgraph of the unsigned graph $G' = (V, E^+ \cup E^-)$.

Lemma 5. *For $p > 0$ and $\alpha > 0$, the function $\Phi_{p,\alpha} : 2^V \to \mathbb{R}$, defined as $\Phi_{p,\alpha}(U) := \sum_{u \in U} e^{p \cdot d_U(u)} - \alpha |U|$, is supermodular.*

Proof. A set function f is supermodular if it has increasing marginal returns: for any $S \subseteq T \subseteq V$ and $z \notin T$,

$$f(T \cup \{z\}) - f(T) \geq f(S \cup \{z\}) - f(S).$$

The function $\Phi_{p,\alpha}$ is the sum of $H_p(S) = \sum_{u \in S} e^{p \cdot d_S(u)}$ and the linear term $-\alpha|S|$. Since every modular function is also supermodular, and the sum of two supermodular functions is supermodular, it suffices to show the supermodularity of $H_p(S) = \sum_{u \in S} e^{p \cdot d_S(u)}$. For any node $u \notin S$, we define its degree in S as $d_S(u) = 0$. The marginal gain $\Delta_p(z; S) = H_p(S \cup \{z\}) - H_p(S)$ is given by:

$$\Delta_p(z; S) = e^{p \cdot d_S(z)} + \sum_{v \in N(z) \cap S} \left(e^{p(d_S(v)+1)} - e^{p \cdot d_S(v)} \right).$$

To compare $\Delta_p(z; T)$ and $\Delta_p(z; S)$, we utilize the convexity of the exponential function.

Observation 1. *Let $g(x) = e^{px}$ with $p > 0$. Since $g(x)$ is convex, the discrete difference $g(x+1) - g(x)$ is non-decreasing with respect to x, i.e. for any $x < x'$, it holds $g(x+1) - g(x) \leq g(x'+1) - g(x')$.*

We analyze the components of the marginal gains for $S \subseteq T$:

1. *Self-Contribution:* Since $S \subseteq T$, $d_T(z) \geq d_S(z)$. Because e^{px} is increasing,
$$e^{p \cdot d_T(z)} \geq e^{p \cdot d_S(z)}.$$
2. *Neighbors' Contribution:* We split the summation for T into neighbors in S and neighbors in $T \setminus S$. For neighbors $v \in N(z) \cap S$, we have $d_T(v) \geq d_S(v)$. By Observation 1:
$$e^{p(d_T(v)+1)} - e^{p \cdot d_T(v)} \geq e^{p(d_S(v)+1)} - e^{p \cdot d_S(v)}.$$
Additionally, any neighbors $v \in N(z) \cap (T \setminus S)$ contribute positive terms to $\Delta_p(z; T)$ which are absent (zero) in $\Delta_p(z; S)$.

Since every term in $\Delta_p(z; T)$ is greater than or equal to the corresponding term in $\Delta_p(z; S)$, we conclude $\Delta_p(z; T) \geq \Delta_p(z; S)$.

□

Maximizing the supermodular function $\Phi_{p,\alpha}$ is equivalent to minimizing the submodular function $-\Phi_{p,\alpha}$. Following (Veldt et al., 2021), one might attempt a binary search on α to find the largest value such that $\Phi_{p,\alpha}(S) \geq 0$, i.e., $f_p(S_{1,2}) \geq \alpha$, using submodular minimization as a black box (Orlin, 2009) at each step. However, this approach does not directly yield a polynomial-time algorithm for general finite $p \neq 0$. The binary search requires a finite number of critical thresholds at which the optimal partition changes; in the DS setting of (Veldt et al., 2021), these thresholds are rational with polynomially bounded denominators, ensuring efficient termination. In our setting, the numerator $\sum_{u \in S_1 \cup S_2} e^{p \cdot d^{\pm}_{S_{1,2}}(u)}$ is real-valued, so distinct critical thresholds may be exponentially close, precluding a polynomial bound on the number of binary search steps. Whether a polynomial-time exact algorithm exists for P-MEAN-2PC at general finite $p \neq 0$ remains an open question.

D Background and Related Work

D.1 Dense(st) subgraph discovery

Finding a dense subgraph in an input graph requires to maximize some notion of density (Lee et al., 2010). Different densities lead to different definitions of a dense subgraph (Fang et al., 2022), e.g., cliques (Hastad, 1996), quasi-cliques (Tsourakakis et al., 2013; Sanei-Mehri et al., 2021), k-cliques (Tsourakakis, 2015). A widely-used density notion is the ratio between number of edges and number of nodes, which is equivalent to half the average degree. The corresponding problem of finding a subgraph that maximizes the average degree is known as the Densest Subgraph (DS) problem.

Let $G = (V, E)$ be an undirected and unweighted graph, with node set V and edge set $E \subseteq V \times V$. Given a node $v \in V$, we denote with $d(v)$ its degree. For a subset $S \subseteq V$ of nodes, $E(S) =$

$\{(u,v) \in E \mid u \in S, v \in S\} = E \cap (S \times S)$ identifies the set of edges joining nodes in S, and $d_S(v)$ is the degree of v in the subgraph $(S, E(S))$ induced by S; if $v \notin S$, then $d_S(v) = 0$. The DS problem is as follows:

Problem 4 (DS (Goldberg, 1984)). *Given a graph $G = (V, E)$, find $S^* \subseteq V$ such that:*

$$S^* = \arg\max_{S \subseteq V} \frac{|E(S)|}{|S|} = \arg\max_{S \subseteq V} \frac{1}{|S|} \sum_{v \in S} d_S(v) \tag{14}$$

DS is solvable in polynomial time using maximum flow (Goldberg, 1984) or linear programming (Charikar, 2003), although they are computationally expensive. A simpler and faster $1/2$-approximation algorithm known as GREEDY or the *peeling algorithm* (Charikar, 2003) iteratively removes nodes with the smallest degree, forming a nested sequence of sets $V = S_n \supset S_{n-1} \supset \cdots \supset S_1 \supset \emptyset$. The set S_j with maximum degree density $|E(S_j)|/|S_j|$ is returned. Recent advances include GREEDY++ (Boob et al., 2020), which enhances GREEDY by running it multiple times and updating node priorities for removal based on previous iterations. Initially conjectured to converge to the optimum, its $(1+\epsilon)$-approximation was later proved by (Chekuri et al., 2022), Fractional peeling (Harb et al., 2022) provides an ϵ-additive approximation, with strong empirical performance. Constrained variants of DS, e.g., the densest k-subgraph, are well-studied, though often NP-hard (Lanciano et al., 2024).

Another well-studied dense subgraph problem is to find the k-core of a graph for a positive integer k (Malliaros et al., 2020). The k-core of a graph is a maximal connected subgraph in which all nodes have degree at least k. The maximum value of k for which the k-core of a graph is non-empty is called the degeneracy of the graph. If k^* is the degeneracy of G, we will refer to the k^*-core of G as the MAXCORE. Finding the MAXCORE is equivalent to the following optimization problem:

Problem 5 (MAXCORE (Seidman, 1983)). *Given a graph $G = (V, E)$, find $S^* \subseteq V$ such that:*

$$S^* = \arg\max_{S \subseteq V} \min_{v \in S} d_S(v) \tag{15}$$

The GREEDY peeling for DS directly yields MAXCORE by tracking the minimum degree during removals: the maximum, over all peeling iterations, of the current minimum degree equals the graph's degeneracy (i.e., the optimum MAXCORE value), and the remaining vertex set at the iteration attaining this maximum is a maximum k-core.

Veldt et al. (Veldt et al., 2021) propose a p-mean densest subgraph objective that unifies the standard DS (when $p = 1$) and MAXCORE (when $p = -\infty$) problems, and provides a general framework for capturing different notions of density in the same graph.

D.2 SIGNED GRAPH CLUSTERING AND SUBGRAPH MINING.

A related line of work studies community detection in signed networks by partitioning all nodes into k communities (Chiang et al., 2012; Cucuringu et al., 2019; He et al., 2022; Kunegis et al., 2010; Mercado et al., 2016). Representative examples include BNC (Chiang et al., 2012), which optimizes balanced normalized cut, SPONGE (Cucuringu et al., 2019), which solves a generalized eigenvalue problem, and SSSNET (He et al., 2022), a semi-supervised GNN model for signed graph partitioning. Chu *et al.* (Chu et al., 2016) instead formulate the search for k oppositive cohesive groups as a constrained quadratic optimization problem. However, these approaches differ fundamentally from P-MEAN-2PC: they enforce a full partition of the graph and evaluate the objective over all nodes, thus forcing even weakly polarized or neutral nodes into some cluster. In contrast, P-MEAN-2PC explicitly admits a neutral set S_0 and optimizes polarity only over the selected polarized nodes S_1 and S_2. This makes it better suited to settings where polarization is concentrated in only part of the network (Bonchi et al., 2019). Other works extract signed subgraph patterns according to objectives different from polarity. TIMBAL finds a maximum-size perfectly balanced subgraph, without explicitly favoring dense solutions (Ordozgoiti et al., 2020). Yao *et al.* (Yao et al., 2022) study maximum balanced clique mining, which searches for the largest structurally balanced clique satisfying side-size constraints; this is a stricter pattern-mining objective, since every pair of selected vertices must be connected and all observed edges must be consistent with a balanced bipartition. Xiao *et al.* (Xiao et al., 2020) optimize signed bipartiteness ratio in a local seed-expansion setting, emphasizing separation from the rest of the graph rather than internal polarized density. Niu and Sariyüce (Niu & Sariyüce, 2023) study dichotomy, a polarity-related notion with stronger emphasis on cohesiveness. Finally, Chu *et al.* (Chu et al., 2016) optimize oppositive cohesiveness, which is

Algorithm 3 Core-2PC-Peel

Input: Signed graph $G = (V, E^+, E^-)$
Output: A pair $\widehat{S}_{1,2} = \{\widehat{S}_1, \widehat{S}_2\}$ of polarized communities

```
S_{1,2}^{SEED} ← EIGEN-BIPARTITION(G) {Algorithm 2}
Ŝ_{1,2} ← S_{1,2}^{SEED},  S_{1,2} ← S_{1,2}^{SEED}
while |S_1 ∪ S_2| > 1 do
    z = arg min_{v∈S_1∪S_2} d^±_{S_{1,2}}(v)
    remove z from S_1 or S_2 in S_{1,2}
    if min_{v∈S_1∪S_2} d^±_{S_{1,2}}(v) > min_{v∈Ŝ_1∪Ŝ_2} d^±_{Ŝ_{1,2}}(v) then
        Ŝ_{1,2} ← S_{1,2}
return Ŝ_{1,2}
```

defined in absolute rather than density-normalized terms. None of these formulations provides a tunable parameter that continuously interpolates between different notions of polarized communities; P-MEAN-2PC fills this gap within the polarity-based framework.

Moreover, our CORE-2PC formulation is also distinct from the (p, n)-core model for signed networks (Kim et al., 2023): that line of work finds maximal subgraphs satisfying fixed local thresholds on positive and negative degrees, while the CORE-2PC problem optimizes the minimum node-wise net degree balance over a pair of polarized communities, without imposing hard per-node thresholds on positive/negative edge counts.

E TIME COMPLEXITY OF GENPEEL-p-2PC

Proposition 4. *GenPeel-p-2PC runs in $\mathcal{O}(|V||E|)$ time.*

Proof. For a signed graph with $|V|$ nodes and $|E|$ edges (including both positive and negative edges), the time complexity of the first step is determined by the cost of finding the largest eigenvector of the signed adjacency matrix (EIGEN-BIPARTITION algorithm) and the subsequent cost of the rounding step. The former step can be done through the Lanczos method in $\mathcal{O}(t \cdot |E|)$, where t is the number of Lanczos iterations (typically constant in the size of the input graph). The rounding step can be done in $\mathcal{O}(|V|)$ time.

The cost of GenPeel-p-2PC is higher than in standard degree-based peeling, since vertices are not removed according to their current net degree balance, but according to the exact marginal loss $\Delta_p(z; S_{1,2})$, i.e., the decrease in the numerator of the $f_p(S_{1,2})$ (cf. Eq. 13) caused by removing z. This quantity depends not only on the current net degree balance of z, but also on the current net degree balances of all its neighbors. As a consequence, after removing a node, the scores of its neighbors (either consistent or inconsistent) change, and so do the Δ_p-values of nodes in its two-hop neighborhood. Removing the node with the smallest (resp. largest) $\Delta_p(z; S_{1,2})$ for $p > 0$ (resp. $p < 0$) can be done by a linear scan in $\mathcal{O}(|V|)$ time.

After removing a vertex z, the algorithm updates the net degree balances of its current neighbors $N(z)$ and recomputes the Δ_p-value only of the vertices in $R(z) = N(z) \cup N(N(z))$, since all other Δ_p-values are unaffected. Given the current net degree balances, computing $\Delta_p(v; S_{1,2})$ for one vertex v requires scanning its current neighbors, and therefore costs $\mathcal{O}(d(v))$ time. Identifying $R(z)$ costs $\sum_{u \in N(z)} \mathcal{O}(d(u)) = \mathcal{O}(|E|)$, and, since each vertex of $R(z)$ is recomputed once, the recomputation costs $\sum_{v \in R(z)} \mathcal{O}(d(v)) = \mathcal{O}(|E|)$. Together with the linear scan, one peeling iteration therefore costs $\mathcal{O}(|V| + |E|)$. Since exactly one vertex is removed per iteration, the peeling phase performs at most $|V|$ iterations, yielding an overall complexity of $\mathcal{O}(|V|(|V| + |E|))$, i.e., $\mathcal{O}(|V||E|)$ whenever $|E| = \Omega(|V|)$.

□

F FURTHER DETAILS ON THE CORE-2PC-PEEL ALGORITHM.

Algorithm 3 reports the pseudocode of the peeling algorithm used for CORE-2PC. Starting from the seed solution produced by EIGEN-BIPARTITION, the algorithm iteratively removes the node with minimum current net degree balance, evaluates the minimum net degree balance of the remaining solution, and finally returns the best solution encountered over the peeling sequence. The proposed Core-2PC-Peel runs in time linear in the size of the input signed graph.

Proposition 6. *Core-2PC-Peel runs in $\mathcal{O}(|V| + |E|)$ time.*

Proof. For a signed graph with $|V|$ nodes and $|E|$ edges (including both positive and negative edges), the time complexity of the first step is determined by the cost of finding the largest eigenvector of the signed adjacency matrix (EIGEN-BIPARTITION algorithm, line 2) and the subsequent cost of the rounding step. The former step can be done through the Lanczos method in $\mathcal{O}(t \cdot |E|)$, where t is the number of Lanczos iterations. The rounding step can be done in $\mathcal{O}(|V|)$ time. The cost of the greedy peeling algorithm in our context can be implemented to run in linear time. For each possible net degree balance $d = -(|V| - 1), \ldots, |V| - 1$, vertices with net degree balance d are stored in a doubly-linked list. During the initial iteration, the algorithm scans these lists in increasing order of d until it identifies a vertex. Once a vertex z is selected, it is removed from the list, and each of its neighbors is moved to the next lower or upper list, depending on whether it was connected to the removed node by a consistent or inconsistent edge. Thanks to the structure of doubly-linked lists, this entire operation can be executed in $\mathcal{O}(d(z))$ time.

In subsequent iterations, it suffices to return to the list immediately below the one of the last removed vertex, as no vertex can exist in levels below that. This means that throughout the algorithm, the scan moves down at most once per iteration, so the total scanning cost is $\mathcal{O}(|V|)$. Consequently, the overall operations can be performed in $\mathcal{O}(|V| + |E|)$ time. Moreover, maintaining the objective value of the current solution is straightforward, since in CORE-2PC it is simply the minimum current net degree balance. Hence, at each iteration, we only need to monitor this minimum and record the iteration at which it increases; the best solution is reconstructed at the end from the removal order in $\mathcal{O}(|V|)$ time. Assuming the number of iterations of the Lanczos method t is constant in the size of the input graph, we can conclude that the overall time complexity is $\mathcal{O}(|V| + |E|)$. □

G DETAILS ON REAL DATASETS

The *Bitcoin* and *Epinions* datasets represent trust-distrust networks from the Bitcoin OTC trading platform and the Epinions social network, respectively. *Cloister* captures esteem and disesteem relationships among monks residing in a cloister in New England (USA). *Congress* records favorable and unfavorable mentions exchanged among politicians speaking in the US Congress. *HTribes* reflects alliances and enmities within a tribe in New Guinea. *Slashdot* is a friend-foe network derived from the Slashdot technology news platform. *TwitterRef* contains tweets related to the Italian constitutional referendum of 2016, where edge signs indicate whether users share the same stance. *WikiCon* includes positive and negative edit conflicts among English Wikipedia users. *WikiEle* represents positive and negative votes cast during the election of English Wikipedia administrators, while *WikiPol* documents signed interactions among users editing English Wikipedia pages on political topics. *Word* encodes synonym (positive edges) and antonym (negative edges) relationships among English words (nodes). The key characteristics of these datasets are summarized in Table 2.

H DETAILS ON M-SSBM

The M-SSBM model is characterized by three key parameters: the total number of nodes, the size $n_c = |S_1| = |S_2|$ of the planted polarized communities (all of equal size), and a noise parameter $\eta \in [0, 1]$ controlling edge probabilities. Specifically: (i) edges within the same group (resp., between polarized groups) are positive (resp., negative) with probability $1 - \eta$, negative (resp., positive) with probability $\eta/2$, and absent with probability $\eta/2$; (ii) all other edges (not part of the polarized communities) are randomly assigned as positive or negative with equal probability $\min(\eta, 1/2)$. As η decreases, the noise level in the graph also decreases. When $\eta = 0$, the graph exhibits a "perfect" polarized structure (nodes are only connected within or between polarized communities), whereas polarized communities are identifiable only when $\eta \leq 2/3$. For $\eta > 2/3$, the graph structure becomes dominated by negative intra-group and positive inter-group edges, making polarization indistinguishable.

In our experiments, following existing works (Bonchi et al., 2019; Tzeng et al., 2020; Gullo et al., 2024; 2026), we generated synthetic graphs with 1000 nodes, community size $n_c = 100$, and varying

Table 2: Main characteristics of real data used in our evaluation. $E = E^+ \cup E^-$: overall edge set; density: $|E|/(|V|(|V|-1)/2)$; $\deg^+$, $\deg^-$: avg of positive and negative node degrees, resp.; cc^+, cc^-: #connected components in the subgraph induced by E^+ and E^-, resp.; cc: overall #connected components.

dataset	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert E^-\rvert/\lvert E\rvert$	density	$\deg^+$	$\deg^-$	cc^+	cc^-	cc
Bitcoin (Leskovec & Krevl, 2014)	5 881	21 492	0.152	0.00124	6.2	1.11	355	4 344	4
Cloister (Kunegis, 2013)	18	125	0.552	0.81699	6.22	7.67	1	1	1
Congress (Kunegis, 2013)	219	521	0.205	0.02183	3.78	0.98	8	127	1
Epinions (Leskovec & Krevl, 2014)	131 580	711 210	0.171	8e-05	8.97	1.84	23 366	90 846	5 568
HTribes (Kunegis, 2013)	16	58	0.5	0.48333	3.62	3.62	2	2	1
Slashdot (Leskovec & Krevl, 2014)	82 140	500 481	0.239	0.00015	9.28	2.91	7427	46 991	1
TwitterRef (Lai et al., 2018)	10 884	251 406	0.051	0.00424	43.85	2.35	69	6 801	11
WikiCon (Leskovec & Krevl, 2014)	116 717	2 026 646	0.628	0.0003	12.9	21.83	70 284	1 788	1 785
WikiEle (Kunegis, 2013)	7 115	100 693	0.221	0.00398	22.05	6.26	892	2 926	24
WikiPol (Lai et al., 2018)	138 587	715 883	0.123	7e-05	9.06	1.27	14 458	97 459	305
Word (Sedoc et al., 2017)	4 962	47 088	0.199	0.00383	15.2	3.77	15	801	1

noise parameter $\eta \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$. For each parameter setting, we produced 10 graphs and reported average results.

I Hardware Configuration

All the experiments were conducted using a double 56-core Intel(R) Xeon(R) Gold 6258R CPU, equipped with 256GB RAM and two NVIDIA GeForce RTX3090s with 24GB memory each, OS Ubuntu Linux 22.04 LTS.

J Normalizing p Across Graphs

As discussed in Section 4, p is a modeling choice that specifies which notion of polarized community is sought. This appendix addresses a complementary question: how the *same* numerical value of p behaves across graphs with different degree and net-balance scales.

The effect of a finite p depends on the scale of the node-level balances, since the exponential mean weights each balance through $e^{\,p\, d^{\pm}_{S_{1,2}}(u)}$. Consequently, the same numerical p may remain close to the average-based regime on a graph with small balances, while placing substantially greater emphasis on extreme values on a graph with larger balances. When a comparable degree of minimum- or maximum-oriented emphasis across graphs is desired, a natural option is to normalize the net degree balances by the maximum unsigned degree Δ_G of G and define the dimensionless parameter $\alpha = p\,\Delta_G$, or equivalently $p_G = \alpha/\Delta_G$. Since $d^{\pm}_{S_{1,2}}(u)/\Delta_G \in [-1, 1]$, a common α controls the strength of the emphasis over the same normalized balance range. Moreover, applying the exponential quasi-arithmetic mean with parameter α to the normalized balances gives

$$\frac{1}{\alpha}\log\left(\frac{1}{|S_1 \cup S_2|}\sum_{u\in S_1\cup S_2} e^{\,\alpha\, d^{\pm}_{S_{1,2}}(u)/\Delta_G}\right) \;=\; \frac{1}{\Delta_G}\, E_{\alpha/\Delta_G}(S_{1,2}),$$

i.e., it coincides with using $p_G = \alpha/\Delta_G$ on the original balances, up to the positive factor $1/\Delta_G$. As Δ_G does not depend on $S_{1,2}$, this rescaling does not change the optimizer. Values close to $\alpha = 0$ remain close to the average-based 2PC regime, while increasingly negative or positive values place progressively greater emphasis on the minimum or the maximum net degree balance, respectively.

We empirically assessed this parameterization on Word ($\Delta_G = 157$) and TwitterRef. ($\Delta_G = 2784$), which have substantially different degree and net-balance scales: the mean unsigned degree is 18.98 ± 16.37 for Word versus 46.20 ± 113.71 for TwitterRef. ($2.4\times$ larger mean, $6.9\times$ larger standard deviation), and the mean net degree balance at the initial EIGEN-BIPARTITION bipartition is 9.87 ± 12.05 versus 39.03 ± 104.63 ($4.0\times$ larger mean, $8.7\times$ larger standard deviation). Table 3 reports the results.

The contrast between marked and unmarked rows makes the role of normalization explicit. Under the common setting $p = \pm 0.001$ used in the main paper, Word corresponds to $\alpha = \pm 0.157$ and

Table 3: Effect of the dimensionless parameter α on Word ($\Delta_G = 157$) and TwitterRef. ($\Delta_G = 2784$). Unmarked rows use $p_G = \alpha/\Delta_G$; rows marked with † are the original runs at $p \in \{-0.001, 0, 0.001\}$, for which $\alpha = p\,\Delta_G$.

dataset	α	p_G	E_0	$\min / \max d^{\pm}_{S_{1,2}}(u)$	$d_{\min}/\Delta_G$	E_0/Δ_G	$d_{\max}/\Delta_G$	agr.	dens.	$\lvert S_1\rvert/\lvert S_2\rvert$
Word	-10	-0.0637	24.81	15/65	.096	.158	.414	.980	.251	42/62
	-3	-0.0191	24.94	14/67	.089	.159	.427	.980	.243	44/64
	-1	-0.00637	25.03	13/67	.083	.159	.427	.979	.238	45/66
	$-0.157^{\dagger}$	-0.001	25.03	13/67	.083	.159	.427	.979	.238	45/66
	$0^{\dagger}$	0	25.03	13/67	.083	.159	.427	.979	.238	45/66
	$0.157^{\dagger}$	0.001	24.96	13/66	.083	.159	.420	.980	.238	45/65
	1	0.00637	24.94	13/69	.083	.159	.439	.978	.221	49/70
	3	0.0191	24.90	11/72	.070	.159	.459	.975	.217	49/73
	10	0.0637	9.75	0/137	.000	.062	.873	.947	.080	95/43
TwitterRef.	-10	-0.00359	173.64	90/498	.032	.062	.179	.996	.297	4/586
	-3	-0.00108	174.48	89/536	.032	.063	.193	.996	.274	4/639
	$-2.784^{\dagger}$	-0.001	174.49	89/537	.032	.063	.193	.996	.273	4/640
	-1	-0.000359	174.64	88/557	.032	.063	.200	.996	.263	4/666
	$0^{\dagger}$	0	174.66	88/567	.032	.063	.204	.996	.259	4/676
	1	0.000359	174.61	86/587	.031	.063	.211	.996	.250	4/700
	$2.784^{\dagger}$	0.001	174.26	81/627	.029	.063	.225	.996	.233	4/750
	3	0.00108	174.22	81/630	.029	.063	.226	.996	.232	4/753
	10	0.00359	72.54	-3/2390	$-.001$	.026	.858	.991	.019	1/3930

Table 4: Supplementary results: p-polarity E_p and running time (seconds) for each dataset and selected values of p.

Metric	p	Bitc.	Cloi.	Cong.	Epin.	HTri.	Slash.	TwRef.	WiCon.	WiEle.	WiPol.	Word
E_p	$-\infty$	19.00	5.00	4.00	121.0	5.00	53.00	103.0	141.0	41.00	51.00	22.00
	-0.01	29.63	7.37	6.66	161.1	5.71	78.46	152.8	173.8	66.58	80.27	24.49
	-0.001	30.46	7.38	6.69	169.7	5.73	82.26	171.0	193.2	71.88	88.50	24.98
	0	30.57	7.45	6.70	171.2	6.18	82.80	174.7	196.6	72.79	90.07	24.96
	$+0.001$	30.67	7.46	6.74	172.7	6.18	83.37	179.2	10150	73.86	1424	25.01
	$+0.01$	150.5	7.47	6.77	2607	6.19	1746	1612	19065	243.7	9465	25.54
	$+\infty$	795.0	17.00	33.00	3558	10.00	2548	2784	20146	1065	10715	157.0
time (s)	$-\infty$	0.72	.003	.009	13.2	.002	11.6	3.29	22.1	1.19	13.5	1.32
	-0.01	6.79	.009	.02	738	.004	371	177	7003	27.7	1355	5.65
	-0.001	6.05	.006	.03	753	.005	344	166	6986	26.3	1360	4.16
	0	0.62	.005	.01	168	.002	70.8	3.23	124	1.48	176	1.69
	$+0.001$	6.64	.010	.09	440	.003	321	148	3037	25.6	810	4.43
	$+0.01$	5.09	.005	.19	470	.003	316	119	3257	24.1	755	4.04
	$+\infty$	0.14	.002	.008	4.57	.001	3.12	1.83	13.2	0.72	5.08	0.28

remains close to the $p = 0$ regime, whereas TwitterRef. corresponds to $\alpha = \pm 2.784$ and nearly matches the $\alpha = \pm 3$ runs: the same raw p does not impose comparable emphasis on the two graphs. Common α values instead produce aligned directional changes at the same normalized strength. Moving from $\alpha = 0$ to $\alpha = +3$, the normalized minimum balance decreases on both Word ($.083 \to .070$) and TwitterRef. ($.032 \to .029$), the normalized maximum increases ($.427 \to .459$ and $.204 \to .226$), and density decreases ($.238 \to .217$ and $.259 \to .232$). At $\alpha = -3$ both solutions become denser and smaller, whereas at $\alpha = 10$ both strongly emphasize the maximum balance, reaching comparable normalized maxima of .873 and .858.

We stress that *normalization does not force different graphs to return identical solutions, which naturally depend on each graph's structure; rather, it makes the strength of the minimum- or maximum-oriented emphasis comparable across graphs*. The common-p grid remains appropriate for examining the interpolation among regimes within a single graph, while $p_G = \alpha/\Delta_G$ with a common dimensionless α provides a principled guideline when cross-graph comparability is required.

K Additional Results

Additional results on real-world networks. Table 4 complements the main-paper discussion by reporting the objective value E_p optimized by each regime, together with running times. The E_p values follow the expected behavior of the exponential-mean objective. For $p = 0$, E_p coincides with the standard polarity E_0 (cf. Table 1 in the main paper), while for $p \to -\infty$ it corresponds to the minimum net degree balance and is therefore substantially lower, as it measures the worst node in

the solution. Conversely, for positive p, and especially for $p \to +\infty$, E_p becomes increasingly dominated by high-balance nodes. This explains the sharp growth of E_p on large graphs: for instance, on WikiConflict it increases from 196.6 at $p = 0$ to 10,150 at $p = 0.001$, 19,065 at $p = 0.01$, and 20,146 at $p \to +\infty$; similarly, on WikiPolitics it grows from 90.07 at $p = 0$ to 1,424, 9,465, and 10,715, respectively. These large values should not be interpreted as higher standard polarity, but rather as evidence that positive p emphasizes hub-like vertices with very large net degree balance. In contrast, for small and medium-size datasets such as Cloister, Congress, Highland Tribes, and Word, E_p changes relatively smoothly across finite values of p.

Containment across values of p. Figure 7 reports the containment ratio $C(S^{(p_1)}, S^{(p_2)})$ averaged over the 11 real-world datasets, i.e., the fraction of the nodes returned at p_1 that also appear in the solution returned at p_2. Being asymmetric, it complements the Jaccard similarity of Figure 2 by revealing which solutions are nested into which. Within the non-positive regime nesting is almost perfect: the $p \to -\infty$ solution is fully contained in those obtained at $p \in \{-0.01, -0.001, 0\}$ ($C = 1.00$ in all three cases), while the reverse containments are 0.67, 0.61 and 0.63, consistently with the substantially smaller size of the core-like solutions (cf. Table 1); similarly, 98% of the $p = 0$ solution is contained in the $p = -0.001$ one. The picture changes for $p > 0$. Monotone growth of the solutions would imply $C(S^{(0)}, S^{(p)}) = 1$ for every $p > 0$, whereas we measure 0.91 at $p = +0.001$, 0.61 at $p = +0.01$ and 0.50 at $p \to +\infty$: the positive regimes do not merely add nodes to the $p = 0$ solution, but discard a substantial fraction of them. In the opposite direction, only 40% and 25% of the $p = +0.01$ and $p \to +\infty$ solutions, respectively, fall inside the $p = 0$ one, while these two regimes remain mutually coherent ($C = 0.79$ and 0.73). Containment therefore confirms that the transition at $p > 0$ is a restructuring of the returned communities rather than a simple expansion.

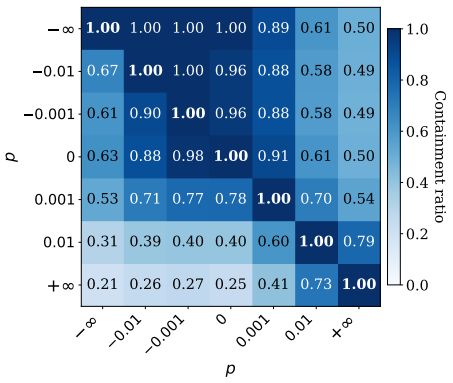


Figure 7: Average containment ratio over the 11 real-world datasets: fraction of the nodes of the row solution that also appear in the column solution.

Additional results on synthetic datasets. Figure 8 reports the F_1-score, the polarity and the agreement ratio as functions of the noise level η, complementing the discussion of **RQ3** in the main paper. Figure 9 decomposes the F_1 trends into precision, recall, and normalized solution size. For finite values of p close to zero, the recovery is almost exact up to moderate noise: for $\eta \leq 0.3$, precision, recall, and F_1 are all equal to 1, and $|S_1 \cup S_2|/(2n_c) = 1$. At $\eta = 0.4$, the same methods remain essentially exact, with precision and recall above 0.995 and normalized size in $[0.996, 1.004]$. At $\eta = 0.5$, the trade-off becomes more visible: positive finite p values, especially $p = 0.01$, favor recall (0.973) by selecting slightly larger communities, whereas negative finite p values are more conservative and yield higher precision.

The extreme regimes clarify the role of the parameter. The core-like solution $p \to -\infty$ is the most selective for $\eta \leq 0.5$: its normalized size decreases from 0.988 to 0.826, while precision remains exactly 1. Hence, its lower F_1 is mainly due to reduced recall, not to the inclusion of noisy vertices. This explains why it consistently achieves the highest agreement ratio for $\eta > 0$. Conversely, $p \to +\infty$ over-expands the solution under noise: its normalized size reaches 2.50 already at $\eta = 0.1$ and 3.657 at $\eta = 0.3$, causing precision to drop to 0.360 and 0.198, respectively. At $\eta = 0.6$, all methods are strongly affected by noise and return oversized solutions, but $p \to -\infty$ remains the most conservative, while $p = 0$ gives the best F_1. Overall, these results confirm that finite p values around zero preserve robust planted-community recovery, whereas the extremes expose meaningful selectivity-versus-expansion trade-offs.

Comparison with additional baselines. For all four baselines we used the implementations released by the original authors: RH (Chen et al., 2024) is written in C++, while TIMBAL (Ordozgoiti et al., 2020), SSSNET (He et al., 2022) and SPONGE (Cucuringu et al., 2019), like our own methods, are in Python. For RH we adopt the iterative variant, which the authors specifically designed for the 2PC problem. SPONGE and SSSNET require the number of clusters k as input; following (Gullo et al., 2026), we run them both with $k = 2$, taking the two resulting clusters as the

Table 5: Comparison with additional baselines on real-world networks. For each dataset we report the average net degree balance E_0, the minimum and maximum net degree balance in the returned solution, the edge-agreement ratio (agr.), the solution edge density (dens.), community sizes $|S_1|/|S_2|$ and running times (in seconds). Boldface marks the best or tied-best value among the reported methods.

criteria	method	Bitc.	Cloi.	Cong.	Epin.	HTri.	Slash.	TwRef.	WCon.	WEle.	WPol.	Word
E_0	$p=-\infty$	28.88	7.17	6.70	140.1	**6.18**	76.39	161.7	163.7	69.53	84.86	22.00
	$p=-0.001$	**30.57**	7.38	6.70	171.1	5.73	**82.80**	174.5	195.4	72.77	90.01	**25.03**
	$p=0$	**30.57**	**7.45**	6.70	**171.2**	**6.18**	**82.80**	**174.7**	**196.6**	**72.79**	**90.07**	24.96
	$p=+0.001$	**30.57**	**7.45**	**6.74**	171.1	**6.18**	82.78	174.3	30.46	72.78	20.62	24.96
	$p=+\infty$	4.39	5.67	3.12	24.43	4.73	7.02	26.06	29.08	24.05	18.05	12.27
	RH	29.30	7.39	6.50	170.5	**6.18**	82.39	174.4	190.5	72.64	89.50	24.28
	SPONGE	8.36	6.11	4.43	7.12	5.50	6.36	28.04	−8.92	15.79	7.79	11.43
	SSSNET	2.34	6.89	4.43	77.06	4.00	7.23	41.50	24.78	17.09	7.20	13.16
	TIMBAL	5.06	6.60	4.36	5.61	5.38	4.00	37.30	16.31	9.96	6.80	9.22
$\min / \max d^{\pm}_{S_{1,2}}(u)$	$p=-\infty$	**19**/69	**5**/9	**4**/16	**121**/148	**5**/8	**53**/121	**103**/347	**141**/314	**41**/275	**51**/230	**22**/22
	$p=-0.001$	16/93	**5**/10	**4**/16	88/262	3/9	42/184	89/537	103/687	37/425	45/405	13/67
	$p=0$	16/93	4/10	**4**/16	86/268	**5**/8	42/184	88/567	102/718	37/452	46/424	13/66
	$p=+0.001$	16/93	4/10	**4**/14	83/273	**5**/8	41/188	81/627	−362/20056	36/457	−26/10390	13/66
	$p=+\infty$	**−4/795**	**−1/17**	**1/33**	**−593/3558**	**1/10**	**−16/2548**	**−448/2784**	**−3277/20146**	**−185/1065**	**−708/10715**	**−1/157**
	RH	1/94	**5**/10	2/17	0/268	1/9	2/184	−1/565	0/1359	18/459	3/442	3/66
	SPONGE	1/19	−1/15	**0/33**	−1090/3088	1/9	−459/2486	−735/1780	−3889/17282	−115/641	−77/5709	−18/100
	SSSNET	0/33	**5**/8	**0/33**	1/546	0/7	−379/2502	−217/2575	−658/19450	−115/657	−159/7031	−8/149
	TIMBAL	1/202	**5**/9	1/31	1/455	3/8	1/1749	1/1674	1/5650	1/392	1/418	1/60
agr.	$p=-\infty$	.972	.939	.995	**1.00**	**1.00**	.995	.997	.982	.933	.971	**1.00**
	$p=-0.001$	.963	.914	.995	**1.00**	.906	.987	.996	.974	.927	.969	.979
	$p=0$	.963	.936	.995	**1.00**	**1.00**	.987	.996	.972	.924	.969	.980
	$p=+0.001$	.963	.936	.995	**1.00**	**1.00**	.987	.996	.825	.924	.914	.980
	$p=+\infty$	.966	.704	**1.00**	.803	.906	.910	.781	.800	.700	.832	.922
	RH	.960	.910	.980	**1.00**	**1.00**	.990	**1.00**	.960	.920	.970	.970
	SPONGE	**1.00**	.720	.965	.829	.879	.761	.803	.372	.779	.877	.801
	SSSNET	**1.00**	.971	.965	.999	.776	.797	.949	.839	.803	.878	.848
	TIMBAL	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
dens.	$p=-\infty$	.262	.742	.109	**.947**	**.618**	**.628**	**.427**	**.529**	**.193**	**.331**	**1.00**
	$p=-0.001$	.174	.744	.109	.651	.505	.423	.273	.183	.124	.186	.238
	$p=0$	.174	.855	.115	.637	**.618**	.423	.259	.178	.116	.176	.238
	$p=+0.001$	.174	.855	**.122**	.625	**.618**	.413	.233	.002	.115	.002	.238
	$p=+\infty$	.006	.817	.094	.011	.582	.003	.017	.002	.056	.003	.093
	RH	.164	.744	.098	.632	.483	.419	.259	.086	.114	.167	.259
	SPONGE	**.310**	.817	.022	8.2e-5	.483	1.5e-4	.0042	3.0e-4	.0040	7.5e-5	.0038
	SSSNET	.0051	**.972**	.022	.086	.483	1.5e-4	.0042	7.9e-4	.0040	7.5e-5	.0038
	TIMBAL	.0013	.733	.021	9.0e-5	.449	1.0e-4	.0042	3.1e-4	.0028	1.1e-4	.0034
$\lvert S_1\rvert/\lvert S_2\rvert$	$p=-\infty$	89/29	9/3	35/28	148/1	7/4	124/0	379/3	221/101	414/2	273/0	14/9
	$p=-0.001$	159/32	10/3	35/28	263/1	11/4	202/0	640/4	825/303	688/2	516/2	66/45
	$p=0$	159/32	8/3	34/26	269/1	7/4	202/0	676/4	865/305	740/2	546/2	65/45
	$p=+0.001$	159/32	8/3	30/27	274/1	7/4	207/0	750/4	18641/1416	745/2	8061/2330	65/45
	$p=+\infty$	786/10	13/5	29/5	3324/235	6/5	2518/31	1540/1245	18715/1432	820/246	8075/2641	107/51
	RH	145/4	10/3	39/33	270/1	7/4	202/1	680/4	1786/590	750/2	572/2	62/38
	SPONGE	26/2	15/3	115/104	131578/2	12/4	82138/2	8274/2610	116712/5	7113/2	138585/2	4954/8
	SSSNET	510/14	6/3	115/104	922/0	11/5	72836/9304	10864/20	51542/4752	6373/742	134086/4501	2925/2037
	TIMBAL	355/3409	7/3	108/98	53481/8878	9/4	35037/4618	8521/417	28405/23607	590/2982	6909/55857	1419/1337
time (s)	$-\infty$	0.72	.003	.009	13.2	.002	11.6	3.29	22.1	1.19	13.5	1.32
	-0.01	6.79	.009	.02	738	.004	371	177	7003	27.7	1355	5.65
	-0.001	6.05	.006	.03	753	.005	344	166	6986	26.3	1360	4.16
	0	0.62	.005	.01	168	.002	70.8	3.23	124	1.48	176	1.69
	$+0.001$	6.64	.010	.09	440	.003	321	148	3037	25.6	810	4.43
	$+0.01$	5.09	.005	.19	470	.003	316	119	3257	24.1	755	4.04
	$+\infty$	0.14	.002	.008	4.57	.001	3.12	1.83	13.2	0.72	5.08	0.28
	RH	6.44	1.04	1.28	41.9	1.04	24.6	17.4	69.2	101	151	10.2
	SPONGE	0.15	.02	.04	1.89	.02	1.64	0.32	2.52	0.20	2.46	4.32
	SSSNET	258	7.30	11.8	7800	6.30	12121	5214	13983	1592	18260	549
	TIMBAL	6.70	6.22	1.12	252	0.81	178	16.0	409	11.0	237	15.2

polarized communities, and with $k = 3$, treating the largest cluster as neutral and the two smaller ones as the polarized communities, and we report for each dataset the best of the two in terms of polarity. Since the selected signed graphs carry no node attributes, SSSNET is fed the final embedding of a signed spectral embedding model (Kunegis et al., 2010) as input feature matrix, with embedding dimension 64, and is trained with the Adam optimizer for 300 epochs, selecting the learning rate from $\{0.01, 0.005, 0.001\}$ and the regularization factor from $\{0.1, 0.01, 0.001\}$ by grid search; for the methods involving randomness, reported values are averages over 30 runs.

Table 5 reports all metrics for all datasets; here we detail how often each method attains the best or tied-best value over the 11 networks. No method is best on every criterion, as different criteria reward different structures. On E_0 the four baselines are rarely competitive: only RH reaches a tied-best value, on a single dataset, whereas the best value is always attained within the P-MEAN-2PC family, by Greedy-2PC ($p = 0$) on 9 datasets and by our algorithms for $p \neq 0$ on 6, the two coinciding on 4. The criteria that $p = 0$ does not target are covered by the regimes designed for them: Max-2PC-Exact attains the best maximum net degree balance on $11/11$, and Core-2PC-Peel the

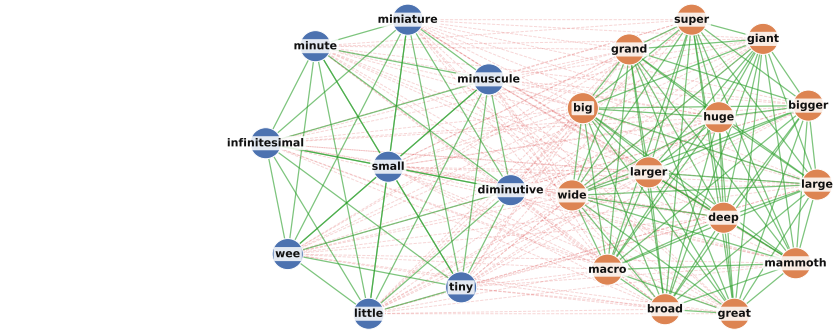
Table 6: Optimization quality on M-SSBM instances: ratio (%) between the objective attained by each method and the exact optimum, as mean ± standard deviation.

		peeling baselines			baselines			
p	GenPeel-p-2PC	Greedy-2PC	Degree-diff	Random	RH	SSSNET	TIMBAL	SPONGE
-0.1	$\mathbf{98.7 \pm 1.8}$	98.2 ± 1.9	70.3 ± 13.0	64.3 ± 10.5	94.0 ± 4.2	82.0 ± 10.0	79.7 ± 13.0	70.0 ± 13.0
-0.05	$\mathbf{98.9 \pm 1.6}$	98.4 ± 1.7	74.6 ± 11.5	69.4 ± 9.3	94.4 ± 3.7	82.5 ± 9.8	79.7 ± 13.0	70.5 ± 12.8
-0.01	$\mathbf{99.1 \pm 1.4}$	98.6 ± 1.6	78.2 ± 10.2	73.7 ± 8.3	94.7 ± 3.4	83.0 ± 9.5	79.8 ± 12.9	71.0 ± 12.5
-0.001	$\mathbf{99.1 \pm 1.3}$	98.6 ± 1.6	79.0 ± 9.9	74.7 ± 8.1	94.8 ± 3.3	83.2 ± 9.5	79.8 ± 12.9	71.2 ± 12.5
$+0.001$	$\mathbf{99.8 \pm 0.4}$	98.6 ± 1.6	79.2 ± 9.9	74.9 ± 8.1	94.8 ± 3.3	83.3 ± 9.5	79.8 ± 12.9	71.3 ± 12.5
$+0.01$	$\mathbf{99.8 \pm 0.4}$	98.6 ± 1.6	80.0 ± 9.6	75.9 ± 7.9	94.8 ± 3.2	83.8 ± 9.3	79.7 ± 12.9	71.8 ± 12.3
$+0.05$	$\mathbf{99.9 \pm 0.3}$	98.6 ± 1.5	83.6 ± 8.2	80.4 ± 6.9	95.0 ± 2.8	85.0 ± 9.0	79.6 ± 12.7	73.0 ± 12.0
$+0.1$	$\mathbf{99.9 \pm 0.1}$	98.6 ± 1.6	88.3 ± 6.5	86.2 ± 5.8	95.3 ± 2.6	86.0 ± 8.8	79.5 ± 12.6	74.0 ± 11.8

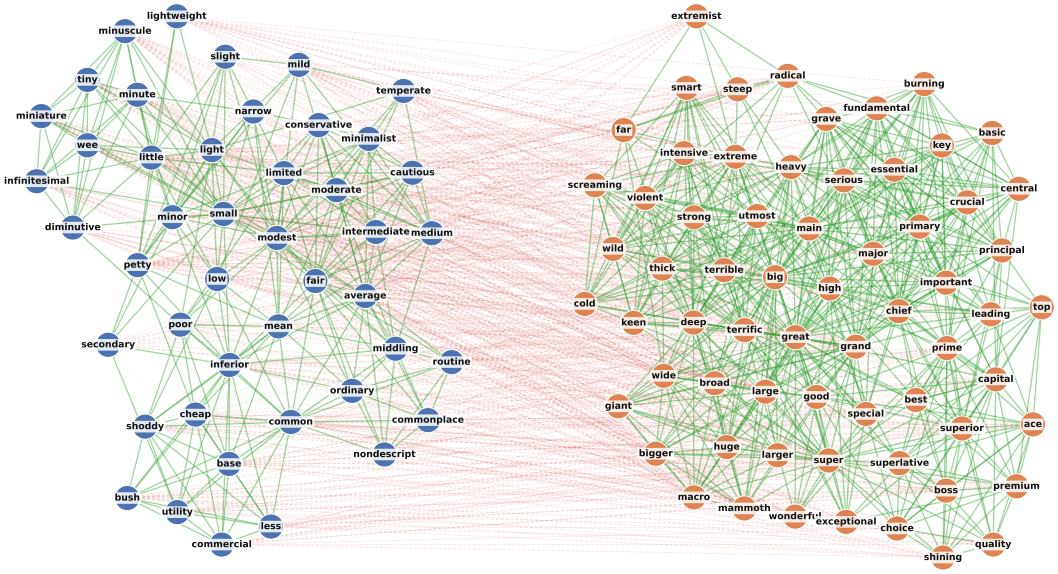
best minimum net degree balance on $11/11$ and the best density on $8/11$. Importantly, Core-2PC-Peel does not trade agreement for cohesion: excluding TIMBAL, whose perfect agreement follows by construction from returning strictly balanced, and typically very sparse, subgraphs, Core-2PC-Peel attains the highest average agreement ratio (.980, versus .969 for RH, the strongest baseline), while exceeding RH in density on $10/11$ datasets. SPONGE and SSSNET, which partition all nodes and therefore include weakly polarized ones, are never best on E_0 and best on the minimum net degree balance on at most one dataset.

Comparison with the exact optimum on small graphs. Since GenPeel-p-2PC has no approximation guarantee, we assess its optimization quality directly, by comparing it with the exact optimum on instances small enough for exhaustive enumeration. We generated 140 M-SSBM instances with 16 nodes and two equally sized planted communities: 5 graphs for each combination of $n_c \in \{2, 3, 4, 5\}$ and $\eta \in \{0, 0.1, \ldots, 0.6\}$. For each instance and each value of $p \in \{-0.1, -0.05, -0.01, -0.001, 0.001, 0.01, 0.05, 0.1\}$, the exact optimum is obtained by enumerating all feasible solutions and selecting the one maximizing $E_p(S_{1,2})$. Besides GenPeel-p-2PC, we consider three peeling rules that share the same EIGEN-BIPARTITION seed and differ only in the removal criterion, so that the comparison isolates the effect of the p-dependent rule of Eq. 6: Greedy-2PC, which removes the node with minimum net degree balance; *Degree-diff*, which removes the node with the smallest raw difference $d^+(u) - d^-(u)$; and *Random*, which removes a uniformly random node. As in GenPeel-p-2PC, each of them returns the best solution encountered along its own peeling sequence according to $E_p(S_{1,2})$. We also run the four baselines of Table 5; since they are not parameterized by p, each returns a single partition, which we score under $E_p(S_{1,2})$ for every p. Table 6 reports $100 \cdot E_p(S_{1,2})(\text{method})/E_p(S_{1,2})(\text{optimum})$, as mean ± standard deviation over all instances.

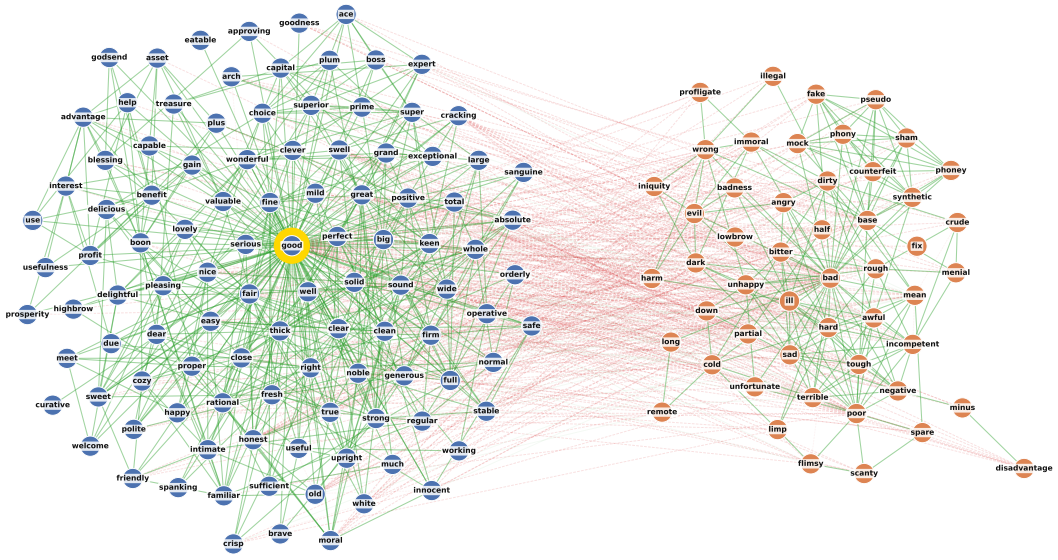
GenPeel-p-2PC attains the highest mean ratio at every value of p, reaching 98.7%–99.9% of the optimum, against 98.2%–98.6% for the Greedy-2PC removal rule; the two simpler rules trail by approximately 10–35 percentage points. The margin over Greedy-2PC, although small in absolute terms, is systematic and grows for $p > 0$, where the cascading effects captured by Eq. 6 are most relevant. Among the baselines, RH is the strongest (94.0%–95.3%, i.e., 4.3–5.0 points below GenPeel-p-2PC), followed by SSSNET, TIMBAL and SPONGE. TIMBAL is essentially flat in p, as expected since it targets exactly balanced subgraphs rather than $E_p(S_{1,2})$: for $p \geq 0.05$ it is overtaken even by *Degree-diff* and *Random*, whose solutions improve as p grows. Overall, GenPeel-p-2PC exhibits near-optimal behavior on this collection of small instances, which, together with the results on real-world networks, provides empirical evidence of good optimization quality in the absence of a general approximation guarantee.



(a) $p = -\infty$

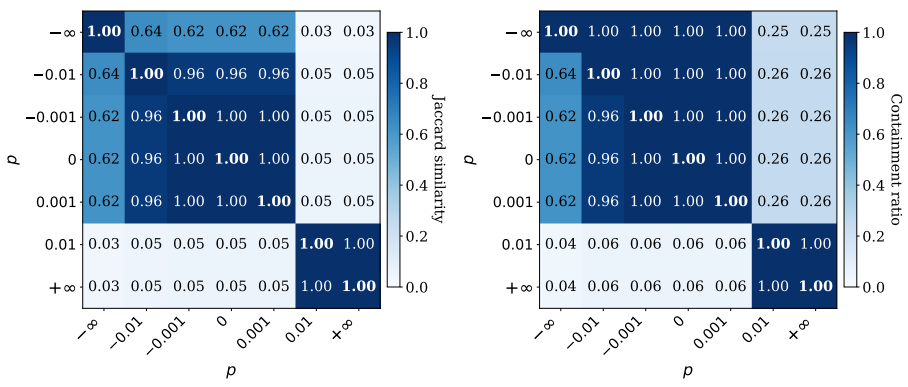


(b) $p = 0$

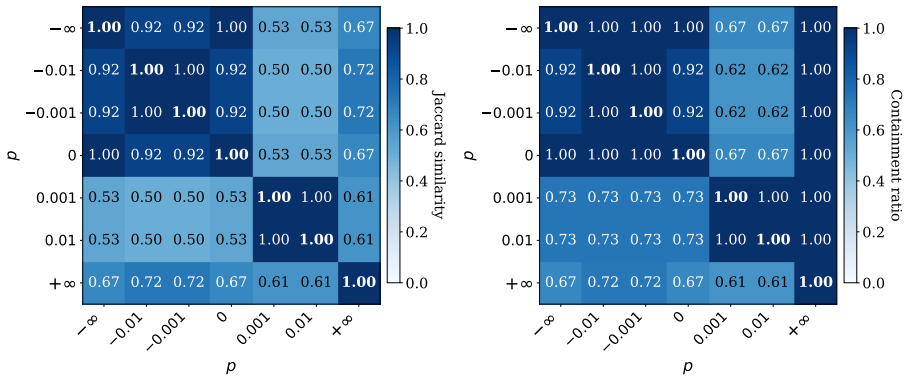


(c) $p = +\infty$

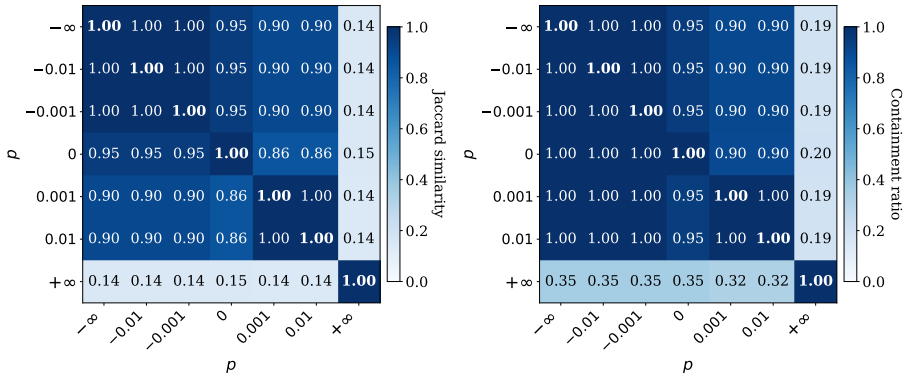
Figure 3: Visualization of the solutions obtained on the Word dataset for $p = -\infty$ (Core-2PC-Peel), $p = 0$ (Greedy-2PC), and $p = +\infty$ (Max-2PC-Exact). In the $p = +\infty$ solution, the yellow circle marks the highest-degree node around which the solution is built.



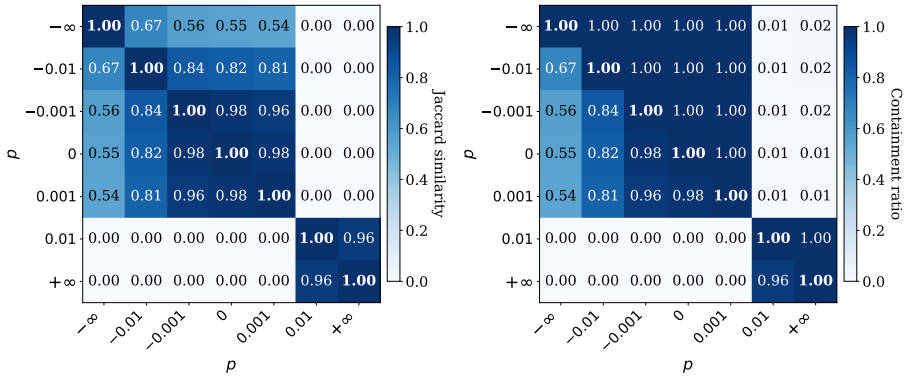
(a) Bitcoin



(b) Cloister

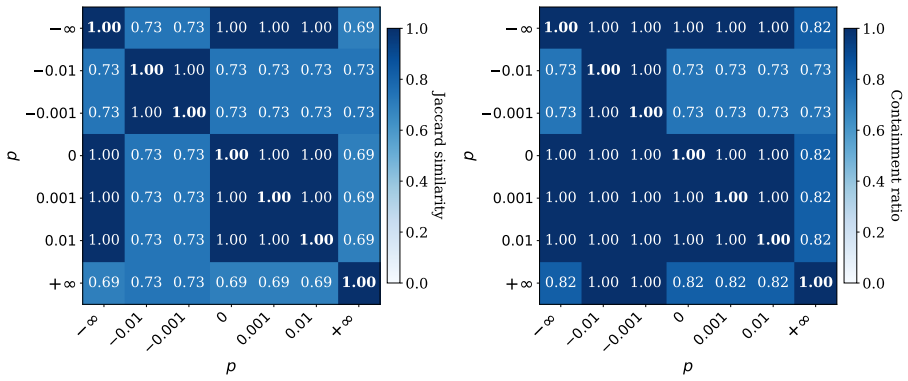


(c) Congress

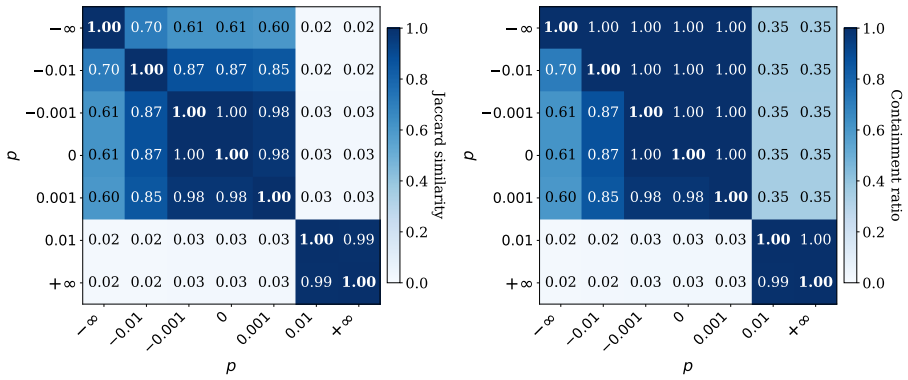


(d) Epinions

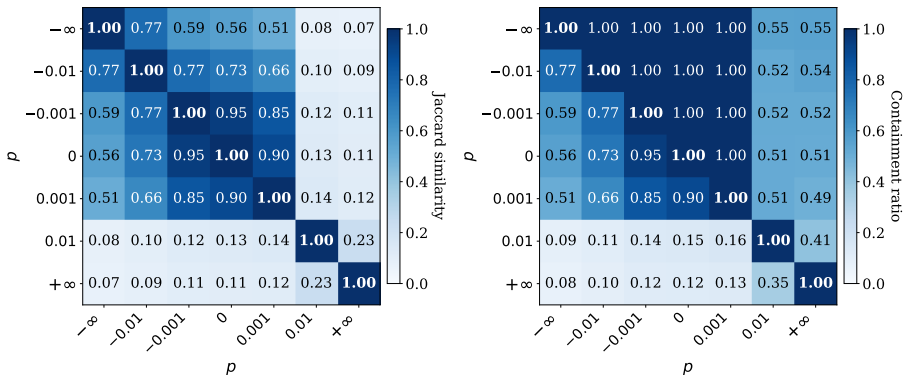
Figure 4: Full per-dataset results on real-world signed networks, part I. Each row corresponds to one dataset, while the two columns report node similarity and containment matrix results, respectively.



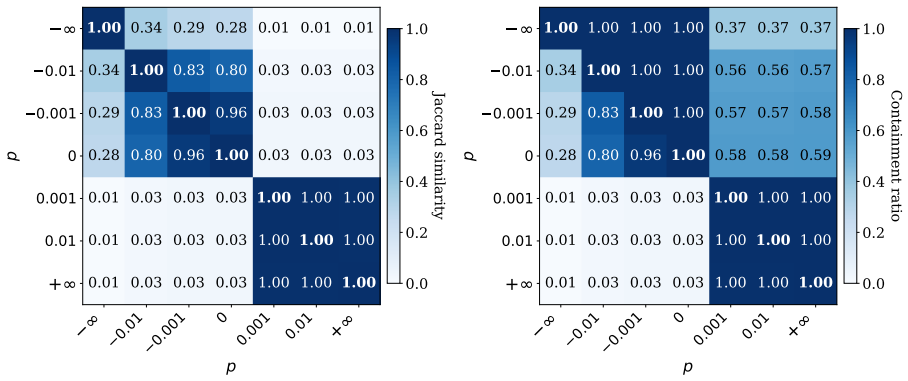
(e) HighlandTribes



(f) Slashdot

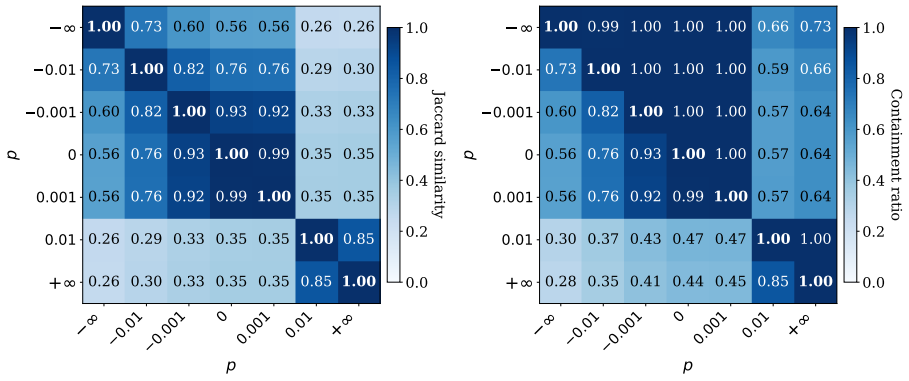


(g) TwitterRef

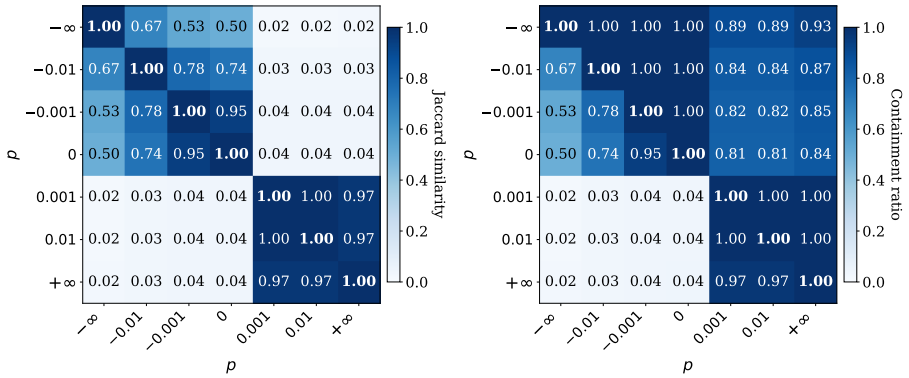


(h) WikiCon

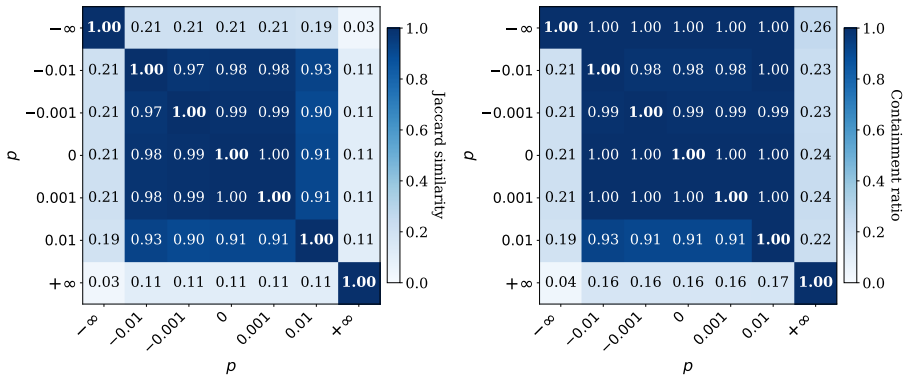
Figure 5: Full per-dataset results on real-world signed networks, part II. Each row corresponds to one dataset, while the two columns report node similarity and containment matrix results, respectively.



(i) WikiEle



(j) WikiPol



(k) Word

Figure 6: Full per-dataset results on real-world signed networks, part III. Each row corresponds to one dataset, while the two columns report node similarity and containment matrix results, respectively.

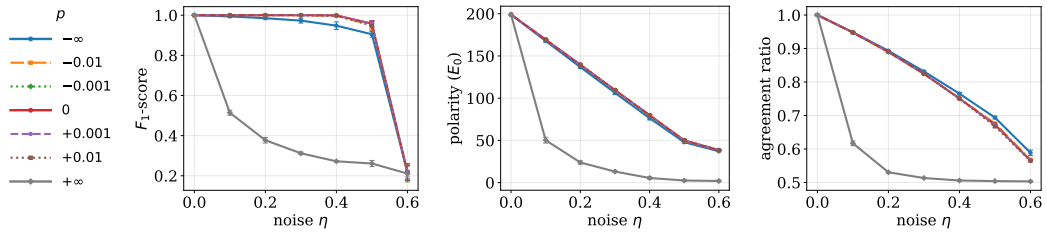


Figure 8: F_1-score (left), polarity score (center), and agreement ratio (right) as functions of the parameter η for the solutions yielded under different values of p.

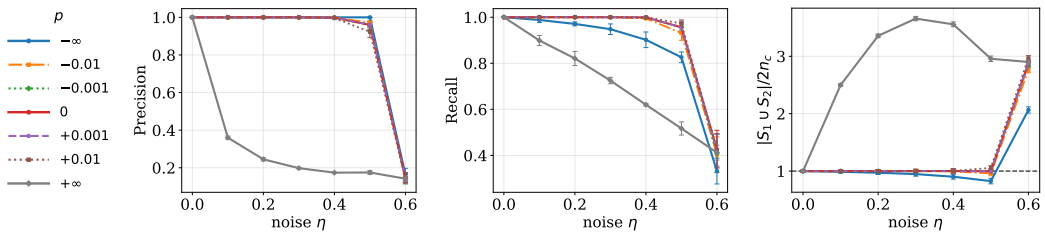


Figure 9: Precision (left), recall (center), and (normalized) size of the yielded communities (right) as functions of the parameter η for the solutions yielded under different values of p.