Proportionality in Ranking Compression

Daniel Halpern
Google Research
dhalpern@google.com

Xinyu Liu
U. of Southern California
xliu4185@usc.edu

Evi Micha
U. of Southern California
evi.micha@usc.edu

Yu Peng Ng
National University of Singapore
ngyupeng@u.nus.edu

Warut Suksompong
National University of Singapore
warut@comp.nus.edu.sg

Abstract

Motivated by applications in reinforcement learning from human feedback, we consider a setting where a given collection of rankings needs to be compressed into a smaller collection of rankings. This can also be seen as a combination of two fundamental social choice frameworks: ranking aggregation and multiwinner voting. We propose three proportionality notions that capture different types of representation in this setting: positional proportionality, pairwise proportionality, and proportionality for solid coalitions (PSC). On the one hand, we show that positional proportionality and PSC are always satisfiable for any input rankings and target compression size, and a desired output can be found in polynomial time. On the other hand, we prove that pairwise proportionality cannot be satisfied in general, but can nevertheless be attained when the input rankings are single-peaked or single-crossing.

1 Introduction

The goal of AI alignment is to build systems that act consistently with human values. For large language models, the predominant method is *reinforcement learning from human feedback* (RLHF) [33]. This process typically consists of three stages: collect user preferences over possible outputs, fit a reward function to best match these choices, and fine-tune the policy to optimize the learned signal. However, standard RLHF implicitly assumes a shared objective among evaluators. When "correct" behavior naturally diverges across cultures or individuals with different backgrounds [2, 27, 36, 44, 45], aggregating these varied viewpoints into a single target marginalizes minority perspectives and leads to *preference collapse* [43], an extreme lack of diversity in outputs [26, 28, 29, 34, 38, 39].

In response to this challenge, the subfield of *pluralistic alignment* has emerged, with an explicit goal of preserving and utilizing diverse belief systems [40]. One practical approach is to learn multiple reward functions from feedback data that collectively cover these diverse views [9, 10, 16, 23, 35]. For example, we could train a distinct model for each reward and deploy them as an ensemble [40]. Alternatively, these rewards or their resulting learned models can act as a "jury" that effectively votes on the final output [22]. In an idealized world, we might learn a unique reward function for every individual in society. While this would cover the full spectrum of views, scaling it to millions (or billions) of personalized rewards is computationally prohibitive. Therefore, to implement any of these ensemble methods, it is necessary to compress the global preferences into a relatively small set of reward functions while preserving their underlying diversity.

To formalize this approach, we turn to the field of *social choice theory*, which studies how to aggregate individual preferences into a collective outcome [4, 7]. In our setting, human annotators naturally provide ranked feedback, and a reward function can be evaluated by how it orders candidate

Preprint.

outputs [21]. Thus, at an abstract level, our problem reduces to the following: given an input profile of n rankings, construct $k \leq n$ output rankings that optimally represent the input.[1] This formalization casts our problem as an intersection of two classic social choice frameworks: *multiwinner voting*, which aims to choose a representative set of candidates, and *rank aggregation*, which concerns selecting a single winning ranking (the setting of Arrow's famous impossibility theorem [3]). By contrast, our setting—which we will refer to as *ranking compression*—requires generating a *representative set of rankings*. In addition to the RLHF application discussed above, our setting also arises, for instance, when community representatives submit rankings to a central aggregation procedure. Naturally, the representatives aim to capture the original rankings of the community members whom they represent as faithfully as possible.

Despite the seemingly simple setup, formalizing the concept of "optimal representation" in this context is far from trivial. Indeed, compressing a large preference profile into a smaller set is inherently lossy, thereby requiring a formal definition of what it means to faithfully reflect the original preferences. Since there is no single canonical choice—and the ideal tradeoff inevitably depends on the downstream task—we instead adopt an axiomatic approach, seeking to establish precisely which properties can or cannot be mathematically preserved under compression.

Our results. We introduce three natural axioms that capture distinct facets of proportional representation in ranking compression.

- *Positional proportionality*: The fraction of input rankings that place a candidate at or above a given rank must be preserved in the output.
- *Pairwise proportionality*: The pairwise comparison between candidates in the input instance must be preserved in the output.
- *Proportionality for solid coalitions (PSC)*: If a group of input rankings ranks a set of candidates above all remaining candidates, this must be reflected in the output. This extends the well-known multiwinner voting axiom (with the same name) to our setting [5, 8, 17].

Our findings offer a holistic analysis of these axioms, demonstrating both existence guarantees (all with accompanying efficient algorithms) as well as impossibility results.

On the positive side, we show that positional proportionality is always achievable for any given input instance and target compression size (Theorem 1). We also prove that the same holds for PSC (Theorem 2). Our proofs for these results leverage various combinatorial and algorithmic tools, including Hall's marriage theorem and network flow.

In contrast, we establish an impossibility for pairwise proportionality, by proving that it cannot necessarily be satisfied for any $k \geq 2$ and $n \geq k + 2$ (Theorem 3). Our counterexamples rely on intricate constructions of constraint graphs and arguments about when the corresponding constraints can be fulfilled by a set of rankings. Despite this negative result, we show that pairwise proportionality can be guaranteed under *single-peaked* or *single-crossing* preferences, two well-studied restricted domains in social choice (Theorems 4 and 5). Moreover, for single-peaked instances we can ensure that the output simultaneously satisfies the other two proportionality notions and preserves single-peakedness, while for single-crossing instances we can select the output from the input rankings themselves in such a way that it also satisfies PSC.

Related work. In terms of problem formulation, our framework shares its basic premise with a "subset selection" problem focused on choosing a representative set of LLM benchmarks from a large pool [37]. That prior work similarly requires outputting k rankings based on an input of n rankings, and defines an approximate version of positional proportionality. However, it crucially demands the output rankings to be drawn directly from the input rankings. By removing this restriction, our setting offers a much larger solution space, thereby necessitating an entirely different set of combinatorial tools. In fact, as we demonstrate in Proposition 1 and Appendix F, most of our positive results cease to hold under the output-from-input restriction.

[1] Admittedly, there are some practical gaps between this theoretical abstraction and traditional RLHF frameworks. For instance, annotators sometimes provide only sparse pairwise comparisons, making it difficult to infer complete preferences (though recent datasets are beginning to collect substantially richer preference data per user [45]). Nevertheless, our primary focus is to establish the theoretical limits of ranking compression; we discuss possible approaches to bridge this divide in Section 6.

Conceptually, ranking compression shares similarities with the clustering of rankings [19, 20, 41]. The general objective of clustering is to partition a preference profile into k disjoint sets, where the k "central" rankings derived from these clusters can be interpreted as a meaningful compression of the profile. However, existing clustering frameworks differ from our approach in several fundamental ways. For example, clusters are typically allowed to vary in size, which suggests that the k representative rankings should be interpreted as weighted. Additionally, a core tenet of traditional clustering is that each voter is explicitly assigned to a specific cluster center. In contrast, our work focuses on achieving a global axiomatic representation without requiring any such individual assignments.

Within the clustering literature, Viappiani [41] examines a standard distance-based clustering method designed to find k rankings that minimize the sum of distances. Recent research has also utilized clustering to measure election polarization [19] and investigated the computational complexity of identifying optimal cluster centers, such as k-Kemeny scores [20]. Further afield are notions of proportional clustering in general metric spaces [6, 11, 14, 25, 32]. While these notions do not normally involve rankings, they can in principle be extended to our setting using ranking-specific metrics such as the Kendall-tau distance or Spearman's footrule.

Finally, while notions of compression exist within social choice theory, they differ significantly from our approach. For instance, *compilation complexity* [12, 13, 42] measures the minimum number of bits needed to encode the votes of a subpopulation without altering the final election outcome. Another related notion is the *communication-distortion tradeoff* [30, 31], which concerns how to efficiently compress real-valued utility functions into fixed-size ballots in order to maximize social welfare.

2 Preliminaries

For any positive integers $z \leq z'$, let $[z] = \{1, \dots, z\}$, $[z]_0 = \{0, 1, \dots, z\}$, and $[z, z'] = \{z, z+1, \dots, z'\}$. Let $N = [n]$ be a set of n voters and $C = [m]$ be a set of m candidates, for some positive integers n, m. Each voter $i \in N$ has a strict ranking $\pi_i = (c_{i,1}, \dots, c_{i,m})$ of the candidates in C. The goal is to compress these n input rankings into an output consisting of k rankings, where $k \in [n]$ is a given target compression size;[2] these k output rankings need not come from the n input rankings, nor must they be distinct. An *instance* of the ranking compression problem is described by a tuple $(N, C, (\pi_i)_{i \in N}, k)$.

We now define three notions of proportionality for the output with respect to the input rankings.

Positional proportionality: Given an instance, an output is said to satisfy *positional proportionality* if the following holds: For any $t \in [k]$ and $\ell \in [m]$, if a candidate appears in the top ℓ ranks in at least $t \cdot n/k$ input rankings, then it must appear in the top ℓ ranks in at least t output rankings.

Pairwise proportionality: Given an instance, an output is said to satisfy *pairwise proportionality* if the following holds: For any $t \in [k]$ and candidates $c, c' \in C$, if c is ranked higher than c' in at least $t \cdot n/k$ input rankings, then c must be ranked higher than c' in at least t output rankings.

PSC: Given an instance, an output is said to satisfy *proportionality for solid coalitions (PSC)* if the following holds: For any $t \in [k]$ and $C' \subseteq C$, if at least $t \cdot n/k$ input rankings have all candidates from C' in the top $|C'|$ ranks (but may order the candidates from C' differently within the top $|C'|$ ranks), then at least t output rankings must have all candidates from C' in the top $|C'|$ ranks.

The following proposition demonstrates that requiring the output rankings to come from the input rankings severely limits the potential for attaining proportionality.

Proposition 1. *Suppose that the output rankings must be drawn from the input rankings. Then, there exists an instance such that none of positional proportionality, pairwise proportionality, and PSC can be attained.*

The full proof of Proposition 1 can be found in Appendix A; we provide a sketch here. Consider an instance with $(n, m, k) = (4, 6, 2)$ and

$$\pi_1 = (3, 4, 2, 5, 1, 6), \quad \pi_2 = (3, 4, 5, 2, 6, 1), \quad \pi_3 = (4, 3, 2, 5, 6, 1), \quad \pi_4 = (4, 3, 5, 2, 1, 6).$$

For positional proportionality, observe that for each pair of input rankings, there exist $c \in C$ and $\ell \in [m]$ such that candidate c appears in the top ℓ ranks *only* within that pair. For example:

[2] While the case $k = n$ is uninteresting since one may simply output the n input rankings, we allow it in our formal model.

- π_1, π_2: candidate 3 appears in the top 1 rank.
- π_1, π_3: candidate 2 appears in the top 3 ranks.
- π_1, π_4: candidate 1 appears in the top 5 ranks.

Analogous statements can be made for the remaining three pairs. Now, each pair imposes a positional constraint on the output that requires selecting at least one of those two rankings. Because no output of size two can intersect all six pairs, positional proportionality cannot be achieved. One can verify that PSC and pairwise proportionality are also unattainable using similar arguments.

3 Positional proportionality

In this section, we consider positional proportionality. We show that an output satisfying positional proportionality always exists regardless of the input rankings and target compression size.

Theorem 1. *For any instance, there exists an output that satisfies positional proportionality. Moreover, such an output can be computed in polynomial time.*

To prove this theorem, we will need the following combinatorial lemma, which can be shown using Hall's marriage theorem. Recall that an $a \times b$ table has a rows and b columns.

Lemma 1. *Let a and b be any positive integers. Given an $a \times b$ table in which each integer $c \in [a]$ appears exactly b times, it is always possible to permute the integers in each row of the table so that in the resulting table, each column forms a permutation of $[a]$.*

The proof of Lemma 1 can be found in Appendix B. We now proceed to prove Theorem 1.

Proof of Theorem 1. We shall construct the output rankings by filling an $m \times k$ table, with each of the k columns corresponding to an output ranking. For any $c \in C$ and $\ell \in [m]$, let $t_{c,\ell}$ be the largest integer such that candidate c appears in the top ℓ ranks in at least $t_{c,\ell} \cdot n/k$ input rankings. It suffices to ensure that c appears in the top ℓ ranks in at least $t_{c,\ell}$ output rankings. For convenience, we additionally let $t_{c,0} = 0$ for all $c \in C$, so that we may define $d_{c,\ell} = t_{c,\ell} - t_{c,\ell-1}$ for all $c \in C$ and $\ell \in [m]$. It is clear that $d_{c,\ell} \geq 0$ for all $c \in C$ and $\ell \in [m]$.

We will fill the table row by row from top to bottom, each row from left to right. Iterating over $j \in [m]$ in increasing order, and then over $c \in C$ in arbitrary order, fill $d_{c,j}$ cells with c. After performing this process up to (and including) the step where $j = \ell$ for some $\ell \in [m]$, the table contains $\sum_{j=1}^{\ell} d_{c,j} = t_{c,\ell} - t_{c,0} = t_{c,\ell}$ occurrences of c. By definition of $t_{c,\ell}$, we have $\sum_{c \in C} t_{c,\ell} \cdot n/k \leq \ell n$, since the top ℓ ranks across the n input rankings consist of ℓn positions in total. Hence, it holds that $\sum_{c \in C} t_{c,\ell} \leq \ell k$. Since we fill the table from top to bottom, all elements that we have filled up to this step are contained within the first ℓ rows of the table. In particular, for any $c \in C$ and $\ell \in [m]$, candidate c appears at least $t_{c,\ell}$ times in the top ℓ rows of the table. Moreover, since $t_{c,m} = k$ for every $c \in C$, each candidate c appears exactly k times in the table.

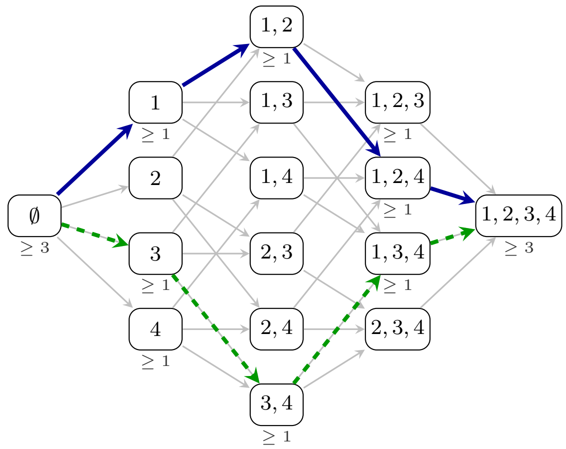
We now apply Lemma 1 to permute the elements in each row of the table so that each column forms a permutation of $[m]$. Since we only permute elements within rows, for any $c \in C$ and $\ell \in [m]$, it still holds that c appears at least $t_{c,\ell}$ times in the top ℓ rows of the table. Therefore, by letting the k columns of the table be the k output rankings, we obtain an output that satisfies positional proportionality.

From this existence proof, we can also derive a polynomial-time algorithm to produce the output. To this end, let $r_{c,\ell}$ be the number of input rankings in which candidate c appears in the top ℓ ranks. First, we compute $r_{c,\ell}$ for all $c \in C$ and $\ell \in [m]$ in $O(nm + m^2)$ time by iterating through the input rankings appropriately. From the values of $r_{c,\ell}$, we compute $t_{c,\ell} = \lfloor r_{c,\ell} \cdot k/n \rfloor$, and then perform our previously described process of filling the $m \times k$ table.

With the $m \times k$ table, we compute $s_{c,\ell}$, the number of times c appears in row ℓ of the table, for all $c \in C$ and $\ell \in [m]$. Following our inductive proof of Lemma 1, we can construct each of the k columns by constructing the described bipartite graph and running the *Hopcroft–Karp algorithm* [24] to find a perfect matching. After constructing each column, we update the counts $s_{c,\ell}$ to construct the subsequent graphs. Since each bipartite graph has m vertices on each of the two sides, it takes $O(m^2)$ time to construct the graph by inspecting the values of $s_{c,\ell}$. Each run of the Hopcroft–Karp

Voters' rankings
$\pi_1 = (1, 2, 4, 3)$
$\pi_2 = (1, 2, 3, 4)$
$\pi_3 = (4, 3, 1, 2)$
$\pi_4 = (3, 4, 1, 2)$
$\pi_5 = (3, 2, 1, 4)$
$\pi_6 = (4, 2, 1, 3)$

(a) Instance with $(n, m, k) = (6, 4, 3)$



(b) Constructed DAG with PSC constraints

Figure 1: An example instance with $(n, m, k) = (6, 4, 3)$ and the corresponding DAG. The constraints below each vertex indicate the lower bound q_S whenever it is nonzero. The solid blue path $\emptyset \to \{1\} \to \{1, 2\} \to \{1, 2, 4\} \to \{1, 2, 3, 4\}$ and the dashed green path $\emptyset \to \{3\} \to \{3, 4\} \to \{1, 3, 4\} \to \{1, 2, 3, 4\}$ correspond to the rankings $(1, 2, 4, 3)$ and $(3, 4, 1, 2)$, respectively.

algorithm takes $O(m^{2.5})$ time [24]. As we construct k bipartite graphs, the overall time complexity is $O(nm + m^2k + m^{2.5}k) = O(nm + m^{2.5}k)$. □

4 Proportionality for solid coalitions

In this section, we consider proportionality for solid coalitions (PSC), which captures a different form of aligned preferences than positional proportionality by focusing on groups of voters with shared top-ranked candidates. We show that, as with positional proportionality, an output satisfying PSC always exists and can be computed in polynomial time.

Before we describe the result, we introduce some necessary notation. For any ranking π over C and any size $\ell \in [m]_0$, let $\mathrm{Top}_\ell(\pi)$ denote the set of the top ℓ candidates in π. For a subset $S \subseteq C$, we define $n_S := |\{i \in N \mid \mathrm{Top}_{|S|}(\pi_i) = S\}|$ as the number of voters who rank the candidates from S in their top $|S|$ positions. The PSC condition requires that for any $S \subseteq C$ and $t \in [k]$, if $n_S \geq t \cdot n/k$, then at least t of the k output rankings have S as their top $|S|$ candidates. Equivalently, for any $S \subseteq C$, at least $q_S := \lfloor n_S \cdot k/n \rfloor$ of the output rankings π must satisfy $\mathrm{Top}_{|S|}(\pi) = S$.

Since PSC depends only on prefixes of rankings, it is helpful to model rankings as chains of growing subsets. To this end, we construct a directed acyclic graph (DAG) $G^C = (V^C, E^C)$ to model the rankings over C. The vertex set $V^C = \mathcal{P}(C)$ consists of all subsets of C. A directed edge $(S, S') \in E^C$ exists if and only if $S \subset S'$ and $|S'| = |S| + 1$. There is a natural bijection between the directed paths from $\emptyset$ to C in G^C and the strict rankings over C. Specifically, a ranking $\pi = (c_1, c_2, \ldots, c_m)$ corresponds to the path $\emptyset = \mathrm{Top}_0(\pi) \to \mathrm{Top}_1(\pi) \to \cdots \to \mathrm{Top}_m(\pi) = C$. In this graph formulation, selecting k rankings satisfying PSC is equivalent to selecting k paths from $\emptyset$ to C such that for every $S \subseteq C$, at least q_S paths visit the vertex S. Figure 1 provides a visual representation of this structure.

It is worth noting that the existence of a valid output does not follow directly from simple greedy heuristics. To see this, consider the instance in Figure 1a with $(n, m, k) = (6, 4, 3)$. A variety of greedy approaches might first produce the following two paths, as illustrated in Figure 1b:

$$\{1\} \to \{1, 2\} \to \{1, 2, 4\} \to \{1, 2, 3, 4\} \quad \text{and} \quad \{3\} \to \{3, 4\} \to \{1, 3, 4\} \to \{1, 2, 3, 4\}.$$

These routing decisions might appear "locally optimal" for this instance because, at every step, each path opportunistically extends to an adjacent vertex with an unmet visit quota. This choice is seemingly justified by the assumption that the vertex must eventually be reached by *some* path. Ultimately, however, these two locally optimal paths cannot be extended to a PSC output for $k = 3$, as the outstanding quotas for vertices $\{4\}$ and $\{1, 2, 3\}$ cannot be satisfied by a single path.

This example shows that purely local strategies are insufficient: constructing a valid path cover requires a more global view of the DAG. In the following theorem, we demonstrate how to obtain such a solution by leveraging classical results from network flow.

Theorem 2. *For any instance, there exists an output that satisfies PSC. Moreover, such an output can be computed in polynomial time.*

Proof. We model the problem with a DAG as described above. Finding a collection of paths that satisfy vertex-visit lower bounds is a classic network flow setup. One can imagine routing k units of flow from $\emptyset$ to C, where each vertex S demands that at least q_S units of flow pass through it. Formally, let $G = (V, E)$ be a directed graph with a *source* $s \in V$ (having no incoming edges) and a *sink* $t \in V$ (having no outgoing edges). Let $L : V \to \mathbb{R}_{\geq 0}$ be a vertex lower bound function. A flow in this network corresponds to an assignment of nonnegative weights to the edges, $f : E \to \mathbb{R}_{\geq 0}$. For any vertex $v \in V$, let $\mathrm{In}(v)$ and $\mathrm{Out}(v)$ denote the sets of its in-neighbors and out-neighbors, respectively. We define the total incoming and outgoing flows at v as $f_{\mathrm{in}}(v) = \sum_{u \in \mathrm{In}(v)} f(u, v)$ and $f_{\mathrm{out}}(v) = \sum_{w \in \mathrm{Out}(v)} f(v, w)$. The total flow passing through a vertex v is given by $f(v) := f_{\mathrm{in}}(v)$ for all $v \in V \setminus \{s\}$, and $f(s) := f_{\mathrm{out}}(s)$. A flow f is considered *valid* if it satisfies two conditions:

- *Flow conservation*: For all intermediate vertices $v \in V \setminus \{s, t\}$, $f_{\mathrm{in}}(v) = f_{\mathrm{out}}(v)$.
- *Demand constraints*: For all $v \in V$, $f(v) \geq L(v)$.

The value of the flow, denoted $\mathrm{val}(f)$, is the net amount of flow leaving the source, $f_{\mathrm{out}}(s)$ (which is equivalently the flow entering the sink, $f_{\mathrm{in}}(t)$).

Standard network flow integrality theorems typically apply to edge upper bounds, but they extend to vertex lower bounds via a known reduction [1]. For completeness, we state this as a lemma, with the proof deferred to Appendix C.

Lemma 2. *Let $G = (V, E)$ be a directed graph with source s and sink t. Let $L : V \to \mathbb{Z}_{\geq 0}$ be an integral vertex lower bound function. If there exists a valid (possibly fractional) flow f satisfying the lower bounds, then there exists a valid integral flow f^* satisfying the same bounds $f^*(v) \geq L(v)$ for all $v \in V$, such that $\mathrm{val}(f^*) \leq \lceil \mathrm{val}(f) \rceil$. Furthermore, f^* can be computed in time $O((|E| + |V|) \sum_{v \in V} L(v))$.*

We apply this framework to G^C with source $s = \emptyset$, sink $t = C$, and integral lower bounds $L(S) = q_S$ for all $S \subseteq C$. We first construct a fractional flow f of value k satisfying these constraints by routing k/n units of flow along each of the n paths corresponding to the input rankings $\pi_1, \dots, \pi_n$. Formally, for each edge $(S, S') \in E^C$, we set

$$f(S, S') = \frac{k}{n} \cdot \left| \left\{ i \in N \;\middle|\; \mathrm{Top}_{|S|}(\pi_i) = S \text{ and } \mathrm{Top}_{|S'|}(\pi_i) = S' \right\} \right|.$$

This flow satisfies conservation and has value k. For any intermediate vertex $S \in \mathcal{P}(C) \setminus \{\emptyset, C\}$, the incoming and outgoing flows correspond exactly to k times the fraction of voters who rank S in their top $|S|$ positions. Thus, $f_{\mathrm{in}}(S) = f_{\mathrm{out}}(S) = n_S \cdot k/n$, which means that $f(S) = n_S \cdot k/n$. For the source and sink, we similarly have $f(\emptyset) = f(C) = n \cdot k/n = k$. The demand constraints are satisfied everywhere because for each $S \subseteq C$,

$$f(S) = n_S \cdot \frac{k}{n} \geq \left\lfloor n_S \cdot \frac{k}{n} \right\rfloor = L(S).$$

Now, by Lemma 2, there exists an integral flow f^* of value $\mathrm{val}(f^*) \leq \lceil \mathrm{val}(f) \rceil = k$ satisfying $f^*(S) \geq L(S)$ for all $S \subseteq C$. Note that any valid flow must have a value of exactly k, since the lower bound constraint on the source requires $\mathrm{val}(f^*) = f^*(\emptyset) \geq L(\emptyset) = \lfloor n \cdot k/n \rfloor = k$.

Because G^C is a DAG and f^* is an integral flow of value k, the flow f^* can be decomposed into k directed paths $P_1, \dots, P_k$ from $\emptyset$ to C. Indeed, this can be done greedily by repeatedly extracting a path from $\emptyset$ to C such that every edge on the path has a positive flow (which must be at least 1), and subtracting 1 from each edge on the path. Since $f^*(S) \geq L(S)$ for all S, at least $L(S) = q_S$ paths visit each vertex S. By interpreting these k paths as rankings, we satisfy the PSC condition.

Finally, we analyze the running time. Note that we do not need to construct the entire DAG G^C (which has 2^m vertices). Instead, we can restrict the network to the subgraph induced by the vertices and edges visited by the paths corresponding to the n input rankings. Since each voter's ranking defines a path of $m+1$ vertices and m edges, this subgraph has at most $n(m+1)$ vertices and nm edges. Furthermore, for any fixed subset size $\ell \in [m]_0$, each voter's path visits exactly one subset S of size ℓ. Therefore, $\sum_{S:|S|=\ell} n_S = n$, which bounds the total demand as $\sum_{S:|S|=\ell} L(S) \leq \sum_{S:|S|=\ell} n_S \cdot k/n = k$. Summing over all $m+1$ subset sizes yields $\sum_{S\subseteq C} L(S) \leq k(m+1) = O(mk)$.

Applying Lemma 2, we can compute the integral flow f^* on this restricted subgraph in $O((|E| + |V|)\sum_S L(S)) = O((nm) \cdot mk) = O(nm^2k)$ time. Extracting the k paths takes $O(nmk)$ time. Thus, the overall process takes $O(nm^2k)$ time. □

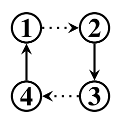
5 Pairwise proportionality

In this section, we turn our attention to pairwise proportionality. While one might intuitively expect that a pairwise proportional output always exists, we prove a surprising impossibility result: for broad ranges of parameters n and k, such an output cannot be guaranteed.

Theorem 3. *For any $k \geq 2$ and $n \geq k+2$, there exists an instance with n voters and target compression size k such that no output satisfies pairwise proportionality.*

We begin by setting up the necessary preliminaries for the case $k = 2$. For each pair of candidates $c, c' \in C$, let $t_{c,c'}$ be the largest integer such that c is ranked higher than c' in at least $t_{c,c'} \cdot n/2$ input rankings. When $t_{c,c'} = 2$, pairwise proportionality imposes the constraint that c must be ranked higher than c' in both output rankings, so we write $c \xrightarrow{\text{s}} c'$ and call this a *strong constraint*. When $t_{c,c'} = 1$, we only need c to be ranked higher than c' in one output ranking, so we write $c \xrightarrow{\text{w}} c'$ and call this a *weak constraint*. We say that a ranking π *satisfies a constraint* $c \to c'$, which can be either strong or weak, if c is ranked higher than c' in π.

We will model the constraints as a directed graph whose vertex set is $C = [m]$. We add a solid edge from c to c' whenever there is a strong constraint $c \xrightarrow{\text{s}} c'$, and add a dotted edge from c to c' whenever there is a weak constraint $c \xrightarrow{\text{w}} c'$. For example, in the figure below, there are strong constraints $4 \xrightarrow{\text{s}} 1$ and $2 \xrightarrow{\text{s}} 3$, and weak constraints $1 \xrightarrow{\text{w}} 2$ and $3 \xrightarrow{\text{w}} 4$. We call the directed graph induced by the constraints a *constraint graph*, and say that a constraint graph is satisfied by two output rankings if all the constraints in the graph are accordingly satisfied by these rankings.



Observe that the constraint graph generated by any input cannot contain a cycle consisting of only strong constraints. Indeed, suppose there is a cycle $c_1 \xrightarrow{\text{s}} c_2 \xrightarrow{\text{s}} \cdots \xrightarrow{\text{s}} c_r \xrightarrow{\text{s}} c_1$. By definition, this means that c_1 must be ranked higher than c_1 in every input ranking, which is impossible.

The following lemma allows us to characterize constraint graphs satisfiable by two output rankings.

Lemma 3. *When $k = 2$, a constraint graph can be satisfied by two output rankings if and only if there exists a ranking π satisfying all strong constraints, such that for every directed cycle in the graph, π satisfies at least one weak constraint in the cycle.*

Proof. ($\Rightarrow$) Suppose that a constraint graph is satisfied by the output rankings π, π'. By definition, π must satisfy all strong constraints. Suppose for contradiction that there exists a directed cycle in the graph such that none of its weak constraints is satisfied by π. This means that every weak constraint in the cycle must be satisfied by π'. Since every strong constraint in the cycle is also satisfied by π', we have that π' satisfies all constraints in the cycle, which is a contradiction.

($\Leftarrow$) Let π be a ranking as described in the lemma statement. We want to construct another ranking π' such that the two output rankings π, π' together satisfy the constraint graph. To this end, it suffices to have π' satisfy all strong constraints in addition to all weak constraints not satisfied by π. We will show that the subgraph induced by this subset of constraints is acyclic, and we can then take π' to be any topological ordering of this subgraph.

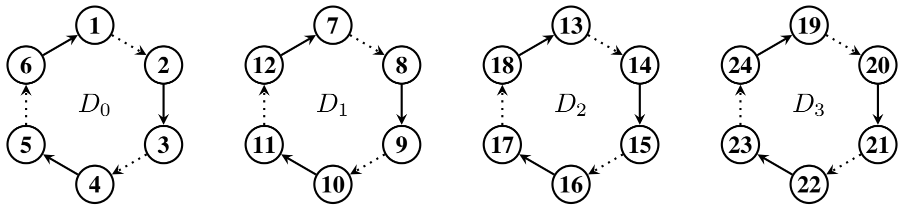


Figure 2: A graph representing part of the constraints in the proof of Theorem 3.

Suppose for contradiction that there exists a directed cycle in this subgraph. Since this cycle also exists in the original constraint graph, π satisfies at least one weak constraint in the cycle, by definition of π. However, this means that this weak constraint should not exist in the subgraph, a contradiction. ☐

We now proceed to prove Theorem 3 for the basic case where $(n, k) = (4, 2)$.

Proof of Theorem 3 for $(n, k) = (4, 2)$. We will specify a constraint graph, show that it cannot be satisfied by any two output rankings, and give four input rankings that generate the constraint graph. These together suffice to complete the proof.

For $j \in [3]_0$, let D_j denote the cycle

$$6j + 1 \xrightarrow{\text{w}} 6j + 2 \xrightarrow{\text{s}} 6j + 3 \xrightarrow{\text{w}} 6j + 4 \xrightarrow{\text{s}} 6j + 5 \xrightarrow{\text{w}} 6j + 6 \xrightarrow{\text{s}} 6j + 1.$$

In particular, there are $m = 24$ candidates. See Figure 2 for an illustration.

For $j \in [3]_0$, denote the sets of odd-numbered and even-numbered vertices in D_j by O_j and E_j, respectively. We will take the indices j of D_j, O_j, E_j modulo 4 hereafter. For each $j \in [3]_0$, we additionally impose a strong constraint $u \xrightarrow{\text{s}} v$ for every pair $(u, v) \in (E_j, O_{j+1})$.

We claim that for every ranking π that satisfies all strong constraints, there exists $j \in [3]_0$ such that none of the weak constraints in D_j is satisfied by π. Suppose for contradiction that there exists a ranking π such that for each $j \in [3]_0$, there exists a weak constraint $u_j \xrightarrow{\text{w}} v_j$ in D_j that is satisfied by π. By our construction, $u_j \in O_j$ and $v_j \in E_j$. This means that there is a strong constraint $v_j \xrightarrow{\text{s}} u_{j+1}$ for each $j \in [3]_0$ as well. Thus, π satisfies all constraints in the cycle

$$u_0 \xrightarrow{\text{w}} v_0 \xrightarrow{\text{s}} u_1 \xrightarrow{\text{w}} v_1 \xrightarrow{\text{s}} u_2 \xrightarrow{\text{w}} v_2 \xrightarrow{\text{s}} u_3 \xrightarrow{\text{w}} v_3 \xrightarrow{\text{s}} u_0,$$

a contradiction. It follows that for every ranking π, there exists a cycle in the constraint graph in which no weak constraint is satisfied by π. By Lemma 3, no two output rankings can satisfy this constraint graph.

It remains to construct four input rankings that generate the constraint graph. For each $j \in [3]_0$ and $j' \in [6]$, let $S_{j,j'}$ be the sequence of vertices that we obtain by traversing the cycle D_j starting from vertex $6j + j'$ and following its edges. For example $S_{2,3}$ is the sequence $15, 16, 17, 18, 13, 14$. We define $\pi_j = (E_j, S_{j+1,6}, S_{j+2,4}, S_{j+3,2}, O_j)$ for $j \in [3]_0$, and take $\pi_0, \pi_1, \pi_2, \pi_3$ as the four input rankings.

For each π_j, we first verify that the strong constraints from $E_{j'}$ to $O_{j'+1}$ for all $j' \in [3]_0$ are satisfied. For $j' = j$, E_j is ranked higher than $S_{j+1,6}$; for $j' = j + 1$, $S_{j+1,6}$ is ranked higher than $S_{j+2,4}$; for $j' = j + 2$, $S_{j+2,4}$ is ranked higher than $S_{j+3,2}$; for $j' = j + 3$, $S_{j+3,2}$ is ranked higher than O_j. Additionally, for all $j' \in [3]_0$ and $j'' \in [3]$, all strong constraints in $D_{j'}$ are satisfied by $S_{j',2j''}$. Therefore, all strong constraints in $D_{j+1}, D_{j+2}, D_{j+3}$ are satisfied by π_j. Moreover, E_j is ranked higher than O_j in π_j, so all strong constraints in D_j are satisfied as well. This means that each π_j satisfies all strong constraints.

Finally, observe for any $j \in [3]_0$, each weak constraint in D_j is satisfied by two of $S_{j,2}, S_{j,4}, S_{j,6}$, which appear in $\pi_{j-3}, \pi_{j-2}, \pi_{j-1}$, respectively. Therefore, the four input rankings together generate all constraints in the graph. This completes the proof. ☐

Our instance in the proof of Theorem 3 requires $m = 24$, which is rather large. Interestingly, an exhaustive search revealed that pairwise proportionality can always be satisfied for $(n, k) = (4, 2)$ up to $m = 7$. This suggests that even though a pairwise proportional outcome is not guaranteed to exist, such an outcome is likely to exist, at least when the parameters are sufficiently small.

The proof for general $k \geq 2$ and $n \geq k + 2$ uses a similar approach as that of the case $(n, k) = (4, 2)$. The full details can be found in Appendix D. We also remark that these bounds on the parameters are tight. Indeed, if $k = 1$ or $n \in \{k, k + 1\}$, we claim that taking any k input rankings as the output already works. This is clear if $k = 1$ or $n = k$. If $n = k + 1$, then whenever there is a constraint that i must be ranked higher than j in at least t output rankings, it must be that i is ranked higher than j in at least $\lceil t \cdot n/k \rceil = \lceil t(k + 1)/k \rceil \geq t + 1$ input rankings. Since we exclude only one input ranking from the output, the constraint is satisfied.

Despite this negative result, we show that pairwise proportionality is always achievable when the input rankings are well-structured. Specifically, this guarantee holds for the two most widely studied classes of structured preferences, *single-peaked* and *single-crossing* [18]; the definitions of these classes are deferred to Appendix E. For single-peaked instances, not only can we find an output satisfying pairwise proportionality, but we can also ensure that the output simultaneously fulfills the other two proportionality notions and is single-peaked with respect to the same candidate ordering. For single-crossing instances, the output can be taken from the input rankings and additionally satisfies PSC.

Theorem 4. *For any single-peaked instance, there exists an output that satisfies pairwise proportionality, positional proportionality, and PSC simultaneously, such that every output ranking is single-peaked with respect to the same ordering of candidates as the input. Moreover, such an output can be computed in polynomial time.*

Theorem 5. *For any single-crossing instance, there exists an output that satisfies pairwise proportionality and PSC simultaneously, such that every output ranking appears in the input. Moreover, such an output can be computed in polynomial time.*

The proofs of Theorems 4 and 5 can be found in Appendix E. While one may hope to add positional proportionality to Theorem 5, we demonstrate in Appendix F that this is in fact impossible, at least without giving up the output-from-input condition. Moreover, since the instance used in our proof of Proposition 1 is single-peaked (see details in Appendix A), adding this condition to Theorem 4 is also infeasible.

6 Future directions

While our paper establishes a solid foundation for understanding proportionality notions in ranking compression, several interesting directions remain to be explored. In particular, we highlight some notable avenues for future work.

Simultaneous proportionality. Our results guarantee the existence of an output satisfying positional proportionality (Theorem 1) and, separately, one satisfying PSC (Theorem 2). Does there always exist an output satisfying both notions simultaneously? In a similar vein, for single-crossing instances (Theorem 5), does there necessarily exist an output satisfying all three notions of proportionality simultaneously, as is the case for single-peaked instances (Theorem 4)? As mentioned at the end of Section 5, such an output cannot always be drawn from the input rankings.

Axiom relaxations. Since pairwise proportionality is not always satisfiable (Theorem 3), can a relaxation of it nevertheless be guaranteed? If we relax the notion multiplicatively (i.e., by requiring candidate c to be ranked higher than candidate c' in at least $r \cdot t \cdot n/k$ *input* rankings), an approximation of $r = 2$ can be achieved. Indeed, we can simply let the output consist of an arbitrary ranking $\lceil k/2 \rceil$ times and the reverse of that ranking $\lfloor k/2 \rfloor$ times. On the other hand, for an additive relaxation (i.e., we require c to be ranked higher than c' in at least $t - r$ *output* rankings), it remains unclear whether an approximation of $r = 1$ is always feasible.

Probabilistic guarantees. Given the adversarial nature of our counterexamples for pairwise proportionality (Theorem 3), is it true that when the input is drawn from certain probabilistic distributions, a pairwise proportional output exists with high probability?

Practical AI alignment. Zooming out, a major challenge lies in translating our theoretical framework into real-world alignment pipelines. For example, can we maintain similar proportionality guarantees if users only provide partial rankings, and what structural assumptions would be required for such data? Finally, deploying this framework entails mapping the output rankings back to tractable reward functions. To ensure that these rewards can be learned in practice, it may be necessary to draw the output rankings from a restricted class that mathematically corresponds to viable functions, similar to the guarantees that we established in the single-peaked setting (Theorem 4).

References

[1] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows: Theory, Algorithms and Applications*. Prentice Hall, 1993.

[2] Usman Anwar, Abulhair Saparov, Javier Rando, Daniel Paleka, Miles Turpin, Peter Hase, Ekdeep Singh Lubana, Erik Jenner, Stephen Casper, Oliver Sourbut, et al. Foundational challenges in assuring alignment and safety of large language models. *Transactions on Machine Learning Research*, 2024.

[3] Kenneth J. Arrow. A difficulty in the concept of social welfare. *Journal of Political Economy*, 58(4):328–346, 1950.

[4] Kenneth J. Arrow, Amartya K. Sen, and Kotaro Suzumura, editors. *Handbook of Social Choice and Welfare*, volume 1. North-Holland, 2002.

[5] Haris Aziz and Barton E. Lee. The expanding approvals rule: Improving proportional representation and monotonicity. *Social Choice and Welfare*, 54(1):1–45, 2020.

[6] Haris Aziz, Barton E. Lee, Sean Morota Chu, and Jeremy Vollen. Proportionally representative clustering. In *Proceedings of the 20th Conference on Web and Internet Economics (WINE)*, pages 155–171, 2024.

[7] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia, editors. *Handbook of Computational Social Choice*. Cambridge University Press, 2016.

[8] Markus Brill and Jannik Peters. Robust and verifiable proportionality axioms for multiwinner voting. In *Proceedings of the 24th ACM Conference on Economics and Computation (EC)*, page 301, 2023.

[9] Souradip Chakraborty, Jiahao Qiu, Hui Yuan, Alec Koppel, Dinesh Manocha, Furong Huang, Amrit Singh Bedi, and Mengdi Wang. MaxMin-RLHF: Alignment with diverse human preferences. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, pages 6116–6135, 2024.

[10] Daiwei Chen, Yi Chen, Aniket Rege, Zhi Wang, and Ramya Korlakai Vinayak. PAL: Sample-efficient personalized reward modeling for pluralistic alignment. In *Proceedings of the 13th International Conference on Learning Representations (ICLR)*, 2025.

[11] Xingyu Chen, Brandon Fain, Liang Lyu, and Kamesh Munagala. Proportionally fair clustering. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pages 1032–1041, 2019.

[12] Yann Chevaleyre, Jérôme Lang, Nicolas Maudet, and Guillaume Ravilly-Abadie. Compiling the votes of a subelectorate. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, pages 97–102, 2009.

[13] Yann Chevaleyre, Jérôme Lang, Nicolas Maudet, and Jérôme Monnot. Compilation and communication protocols for voting rules with a dynamic set of candidates. In *Proceedings of the 13th Conference on Theoretical Aspects of Rationality and Knowledge (TARK)*, pages 153–160, 2011.

[14] Anshuman Chhabra, Karina Masalkovaitė, and Prasant Mohapatra. An overview of fairness in clustering. *IEEE Access*, 9:130698–130720, 2021.

[15] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, 4th edition, 2022.

[16] Thomas Coste, Usman Anwar, Robert Kirk, and David Krueger. Reward model ensembles help mitigate overoptimization. In *Proceedings of the 12th International Conference on Learning Representations (ICLR)*, 2024.

[17] Michael Dummett. *Voting Procedures*. Oxford University Press, 1984.

[18] Edith Elkind, Martin Lackner, and Dominik Peters. Structured preferences. In Ulle Endriss, editor, *Trends in Computational Social Choice*, chapter 10, pages 187–207. AI Access, 2017.

[19] Piotr Faliszewski, Andrzej Kaczmarczyk, Krzysztof Sornat, Stanisław Szufa, and Tomasz Wąs. Diversity, agreement, and polarization in elections. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2684–2692, 2023.

[20] Piotr Faliszewski, Krzysztof Sornat, Stanisław Szufa, and Tomasz Wąs. Diversity of structured domains via k-kemeny scores. In *Proceedings of the 40th AAAI Conference on Artificial Intelligence (AAAI)*, pages 16880–16888, 2026.

[21] Luise Ge, Daniel Halpern, Evi Micha, Ariel D. Procaccia, Itai Shapira, Yevgeniy Vorobeychik, and Junlin Wu. Axioms for AI alignment from human feedback. In *Proceedings of the 38th Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

[22] Mitchell L. Gordon, Michelle S. Lam, Joon Sung Park, Kayur Patel, Jeffrey T. Hancock, Tatsunori Hashimoto, and Michael S. Bernstein. Jury learning: Integrating dissenting voices into machine learning models. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI)*, pages 115:1–115:19, 2022.

[23] Daniel Halpern, Evi Micha, Ariel D. Procaccia, and Itai Shapira. Pairwise calibrated rewards for pluralistic alignment. In *Proceedings of the 39th Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

[24] John E. Hopcroft and Richard M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2(4):225–231, 1973.

[25] Leon Kellerhals and Jannik Peters. Proportional fairness in clustering: A social choice perspective. In *Proceedings of the 38th Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

[26] Muhammad Khalifa, Hady Elsahar, and Marc Dymetman. A distributional approach to controlled text generation. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, 2021.

[27] Hannah Rose Kirk, Alexander Whitefield, Paul Röttger, Andrew Bean, Katerina Margatina, Juan Ciro, Rafael Mosquera, Max Bartolo, Adina Williams, He He, et al. The PRISM alignment dataset: What participatory, representative and individualised human feedback reveals about the subjective and multicultural alignment of large language models. In *Proceedings of the 38th Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

[28] Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the effects of RLHF on LLM generalisation and diversity. In *Proceedings of the 12th International Conference on Learning Representations (ICLR)*, 2024.

[29] Thom Lake, Eunsol Choi, and Greg Durrett. From distributional to Overton pluralism: Investigating large language model alignment. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 6794–6814, 2025.

[30] Debmalya Mandal, Ariel D. Procaccia, Nisarg Shah, and David P. Woodruff. Efficient and thrifty voting by any means necessary. In *Proceedings of the 33rd Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 7178–7189, 2019.

[31] Debmalya Mandal, Nisarg Shah, and David P. Woodruff. Optimal communication-distortion tradeoff in voting. In *Proceedings of the 21st ACM Conference on Economics and Computation (EC)*, pages 795–813, 2020.

[32] Evi Micha and Nisarg Shah. Proportionally fair clustering revisited. In *Proceedings of the 47th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 85:1–85:16, 2020.

[33] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Proceedings of the 36th Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[34] Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. Red teaming language models with language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3419–3448, 2022.

[35] Sriyash Poddar, Yanming Wan, Hamish Ivison, Abhishek Gupta, and Natasha Jaques. Personalizing reinforcement learning from human feedback with variational preference learning. In *Proceedings of the 38th Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

[36] Vinodkumar Prabhakaran, Rida Qadri, and Ben Hutchinson. Cultural incongruencies in artificial intelligence. *arXiv preprint arXiv:2211.13069*, 2022.

[37] Ariel D. Procaccia, Benjamin Schiffer, Serena Wang, and Shirley Zhang. Metritocracy: Representative metrics for lite benchmarks. In *Proceedings of the 39th Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

[38] Shibani Santurkar, Esin Durmus, Faisal Ladhak, Cinoo Lee, Percy Liang, and Tatsunori Hashimoto. Whose opinions do language models reflect? In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, pages 29971–30004, 2023.

[39] Alexander Shypula, Shuo Li, Botong Zhang, Vishakh Padmakumar, Kayo Yin, and Osbert Bastani. Evaluating the diversity and quality of LLM generated content. In *Proceedings of the 1st Conference on Language Modeling (COLM)*, 2025.

[40] Taylor Sorensen, Jared Moore, Jillian Fisher, Mitchell L. Gordon, Niloofar Mireshghallah, Christopher Michael Rytting, Andre Ye, Liwei Jiang, Ximing Lu, Nouha Dziri, Tim Althoff, and Yejin Choi. Position: A roadmap to pluralistic alignment. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, pages 46280–46302, 2024.

[41] Paolo Viappiani. Characterization of scoring rules with distances: Application to the clustering of rankings. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 104–110, 2015.

[42] Lirong Xia and Vincent Conitzer. Compilation complexity of common voting rules. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI)*, pages 915–920, 2010.

[43] Jiancong Xiao, Ziniu Li, Xingyu Xie, Emily Getzen, Cong Fang, Qi Long, and Weijie J. Su. On the algorithmic bias of aligning large language models with RLHF: Preference collapse and matching regularization. *Journal of the American Statistical Association*, 120(552):2154–2164, 2025.

[44] Jiahao Yuan, Zixiang Di, Shangzixin Zhao, and Usman Naseem. Cultural palette: Pluralising culture alignment via multi-agent palette. *arXiv preprint arXiv:2412.11167*, 2024.

[45] Lily Hong Zhang, Smitha Milli, Karen Jusko, Jonathan Smith, Brandon Amos, Wassim Bouaziz, Manon Revel, Jack Kussman, Yasha Sheynin, Lisa Titus, et al. Cultivating pluralism in algorithmic monoculture: The community alignment dataset. In *Proceedings of the 14th International Conference on Learning Representations (ICLR)*, 2026.

A Proof of Proposition 1

Consider the instance below with $(n, m, k) = (4, 6, 2)$.

$$\pi_1 = (3, 4, 2, 5, 1, 6);$$
$$\pi_2 = (3, 4, 5, 2, 6, 1);$$
$$\pi_3 = (4, 3, 2, 5, 6, 1);$$
$$\pi_4 = (4, 3, 5, 2, 1, 6).$$

Note that this instance is single-peaked[3] with respect to the ordering $(1, 2, \ldots, 6)$.

We first check that positional proportionality cannot be achieved by choosing from the input. For each pair of input rankings, there exist $c \in C$ and $\ell \in [m]$ such that candidate c appears in the top ℓ ranks *only* within that pair.

- π_1, π_2: candidate 3 appears in the top 1 rank.
- π_3, π_4: candidate 4 appears in the top 1 rank.
- π_1, π_3: candidate 2 appears in the top 3 ranks.
- π_2, π_4: candidate 5 appears in the top 3 ranks.
- π_1, π_4: candidate 1 appears in the top 5 ranks.
- π_2, π_3: candidate 6 appears in the top 5 ranks.

Each pair imposes a positional constraint on the output that requires selecting at least one of those two rankings. Because no output of size two can intersect all six pairs, positional proportionality cannot be achieved.

In a similar vein, we can show that pairwise proportionality cannot be achieved by choosing from the input rankings. Each pair of rankings is uniquely responsible for a pairwise constraint.

- π_1, π_2: candidate 3 is ranked higher than candidate 4.
- π_3, π_4: candidate 4 is ranked higher than candidate 3.
- π_1, π_3: candidate 2 is ranked higher than candidate 5.
- π_2, π_4: candidate 5 is ranked higher than candidate 2.
- π_1, π_4: candidate 1 is ranked higher than candidate 6.
- π_2, π_3: candidate 6 is ranked higher than candidate 1.

Finally, we address PSC. For each pair of input rankings, we identify a solid coalition that is unique to that pair.

- π_1, π_2: $\{3\}$.
- π_3, π_4: $\{4\}$.
- π_1, π_3: $\{2, 3, 4\}$.
- π_2, π_4: $\{3, 4, 5\}$.
- π_1, π_4: $\{1, 2, 3, 4, 5\}$.
- π_2, π_3: $\{2, 3, 4, 5, 6\}$.

B Proof of Lemma 1

We proceed by induction on b. For the base case $b = 1$, the statement holds trivially. For the inductive step, assume that for some $b \geq 2$, the statement holds for $b - 1$; we will show that it also holds for b. Consider any $a \times b$ table as in the lemma statement. It suffices to show that we can permute each row of the table so that the first column forms a permutation of $[a]$. Indeed, this is because in the

[3] See the definition in Appendix E.

remaining $a \times (b-1)$ table that excludes the first column, every integer $c \in [a]$ appears exactly $b-1$ times, so by invoking the induction hypothesis, we can permute each row of this table so that each column forms a permutation of $[a]$.

To prove our claim, we use Hall's marriage theorem. Construct a bipartite graph $G = (V_L \cup V_R, E)$ where $V_L = \{u_L \mid u \in [a]\}$ represents the integers, $V_R = \{v_R \mid v \in [a]\}$ represents the rows, and an edge $(u_L, v_R) \in E$ exists if and only if row v contains the integer u. We will show that G has a perfect matching. This perfect matching pairs each integer $c \in [a]$ with a unique row containing it. Consequently, we can permute the entries of each row so that every integer c becomes the first entry of its matched row.

To show that a perfect matching exists, it suffices to check that the marriage condition holds. That is, for each set $S \subseteq [a]$ of integers, we need to show that there are at least $|S|$ rows that contain at least one integer from S. Since the integers in S appear exactly $b|S|$ times in the table in total, and each row only contains b integers, these $b|S|$ appearances occur in at least $|S|$ rows. Hence, the marriage condition is satisfied, as required.

C Proof of Lemma 2

We reduce the problem of vertex lower bounds to a standard maximum flow problem with edge capacities. Recall that in a standard flow network with integral capacities, if a valid flow of value F exists, algorithms such as Ford–Fulkerson can find an integral maximum flow in $O(|E| \cdot F)$ time [15]. The reduction involves two standard techniques: using "vertex splitting" to go from vertex constraints to edge constraints, and using a "super-source" and "super-sink" to convert from lower bounds to upper bounds [1].

Auxiliary graph construction. Let $G = (V, E)$ be the original graph with source s, sink t, and integral vertex lower bound function L. Let f be the given valid flow satisfying these constraints.

We construct an auxiliary directed graph $G' = (V', E')$. For every vertex $v \in V$, we introduce two vertices: an "in" vertex v_{in} and an "out" vertex v_{out}. We also add a new source s^* and a new sink t^*. Thus, $V' = \bigcup_{v \in V} \{v_{\text{in}}, v_{\text{out}}\} \cup \{s^*, t^*\}$. The edge set E' and the upper capacity function c are defined as follows:

- *Original edges:* For each $(u, v) \in E$, add an edge $(u_{\text{out}}, v_{\text{in}})$ with capacity $c(u_{\text{out}}, v_{\text{in}}) = \infty$.
- *Internal edges:* For each $v \in V$, add an edge $(v_{\text{in}}, v_{\text{out}})$ with capacity $c(v_{\text{in}}, v_{\text{out}}) = \infty$.
- *Lower bound demands:* For each $v \in V$, add an edge (s^*, v_{out}) with capacity $c(s^*, v_{\text{out}}) = L(v)$, and an edge (v_{in}, t^*) with capacity $c(v_{\text{in}}, t^*) = L(v)$.
- *Circulation edge:* Add a return edge $(t_{\text{out}}, s_{\text{in}})$ from the original sink to the original source, with capacity $c(t_{\text{out}}, s_{\text{in}}) = \lceil \text{val}(f) \rceil$.

Fractional feasibility. The maximum flow from s^* to t^* is upper-bounded by $\sum_{v \in V} L(v)$, which is the capacity of the cut $(\{s^*\}, V' \setminus \{s^*\})$. We show that this maximum is achievable by constructing from f a valid fractional flow f' on G':

$$\begin{aligned} f'(u_{\text{out}}, v_{\text{in}}) &= f(u, v) && \text{for all } (u, v) \in E; \\ f'(v_{\text{in}}, v_{\text{out}}) &= f(v) - L(v) && \text{for all } v \in V; \\ f'(s^*, v_{\text{out}}) &= L(v) && \text{for all } v \in V; \\ f'(v_{\text{in}}, t^*) &= L(v) && \text{for all } v \in V; \\ f'(t_{\text{out}}, s_{\text{in}}) &= \text{val}(f). \end{aligned}$$

We have $f'(v_{\text{in}}, v_{\text{out}}) \geq 0$ because the original flow satisfies the vertex lower bounds $f(v) \geq L(v)$. It is straightforward to verify that all upper capacity constraints are satisfied: the original and internal edges have infinite capacity, the demand edges connected to s^* and t^* exactly match their capacities, and the flow on the circulation edge satisfies $\text{val}(f) \leq \lceil \text{val}(f) \rceil$. Since the edges leaving s^* and entering t^* are saturated, the net flow out of s^* is exactly $\sum_{v \in V} L(v)$.

We verify flow conservation at the internal vertices of G' in two cases:

- **Vertex v_{in} for $v \in V$:** If $v \neq s$, the incoming flow comes from the edges $(u_{\text{out}}, v_{\text{in}})$ for $u \in \text{In}(v)$, summing to $f_{\text{in}}(v) = f(v)$. If $v = s$, the incoming flow comes solely from the circulation edge

$(t_{\text{out}}, s_{\text{in}})$, which carries $\text{val}(f) = f(s)$. Thus, the incoming flow is always $f(v)$. The outgoing flow splits between (v_{in}, t^*) carrying $L(v)$ and $(v_{\text{in}}, v_{\text{out}})$ carrying $f(v) - L(v)$, which sums to $f(v)$. Therefore, conservation holds.

- **Vertex v_{out} for $v \in V$:** The incoming flow comes from s^* carrying $L(v)$ and from v_{in} carrying $f(v) - L(v)$, summing to $f(v)$. For the outgoing flow, if $v \neq t$, it is routed to $(v_{\text{out}}, u_{\text{in}})$ for $u \in \text{Out}(v)$, summing to $f_{\text{out}}(v) = f(v)$. If $v = t$, it is routed solely to the circulation edge $(t_{\text{out}}, s_{\text{in}})$, which carries $\text{val}(f) = f(t)$. Thus, the outgoing flow is always $f(v)$. Therefore, conservation holds.

Integrality and verification. Since a flow of value $\sum_{v \in V} L(v)$ is achievable, and this matches the capacity of the minimum cut $(\{s^*\}, V' \setminus \{s^*\})$, it is a maximum flow. Because all capacities in G' are integral, the Ford–Fulkerson algorithm is guaranteed to find an integral flow g^* on G' with this same maximum value.

Since g^* is a maximum flow, it must saturate all edges leaving s^* and entering t^*. Thus, $g^*(s^*, v_{\text{out}}) = g^*(v_{\text{in}}, t^*) = L(v)$ for all $v \in V$. We translate g^* back to an integral flow f^* on the original graph G by setting $f^*(u, v) = g^*(u_{\text{out}}, v_{\text{in}})$ for all $(u, v) \in E$. We can check that f^* satisfies all required properties:

(1) **Flow conservation:** For any $v \in V \setminus \{s, t\}$, the incoming flow is $f^*_{\text{in}}(v) = \sum_{u \in \text{In}(v)} g^*(u_{\text{out}}, v_{\text{in}})$. By conservation at v_{in} in G', this equals $g^*(v_{\text{in}}, v_{\text{out}}) + g^*(v_{\text{in}}, t^*) = g^*(v_{\text{in}}, v_{\text{out}}) + L(v)$. Similarly, $f^*_{\text{out}}(v) = \sum_{w \in \text{Out}(v)} g^*(v_{\text{out}}, w_{\text{in}})$, which by conservation at v_{out} in G' equals $g^*(v_{\text{in}}, v_{\text{out}}) + g^*(s^*, v_{\text{out}}) = g^*(v_{\text{in}}, v_{\text{out}}) + L(v)$. Thus, $f^*_{\text{in}}(v) = f^*_{\text{out}}(v)$.

(2) **Vertex lower bounds:** For any intermediate vertex $v \in V \setminus \{s, t\}$, the flow through v is $f^*(v) = f^*_{\text{in}}(v) = g^*(v_{\text{in}}, v_{\text{out}}) + L(v)$. Since flow values are nonnegative, $f^*(v) \geq L(v)$. For the source s, by a similar reasoning as in the previous paragraph, we have $f^*(s) = f^*_{\text{out}}(s) = g^*(s_{\text{in}}, s_{\text{out}}) + L(s) \geq L(s)$. Similarly, for the sink t, we have $f^*(t) = f^*_{\text{in}}(t) = g^*(t_{\text{in}}, t_{\text{out}}) + L(t) \geq L(t)$. Therefore, all demand constraints are satisfied.

(3) **Value constraint:** We claim that the net value of f^* from s to t is exactly the flow on the circulation edge $(t_{\text{out}}, s_{\text{in}})$ in G'. To see this, note that $f^*_{\text{out}}(s) = \sum_{w \in N_{\text{out}}(s)} g^*(s_{\text{out}}, w_{\text{in}})$. By flow conservation at s_{out} in G', this equals $g^*(s_{\text{in}}, s_{\text{out}}) + g^*(s^*, s_{\text{out}}) = g^*(s_{\text{in}}, s_{\text{out}}) + L(s)$. Similarly, by conservation at s_{in}, the incoming flow $g^*(t_{\text{out}}, s_{\text{in}})$ must equal the outgoing flow $g^*(s_{\text{in}}, s_{\text{out}}) + g^*(s_{\text{in}}, t^*) = g^*(s_{\text{in}}, s_{\text{out}}) + L(s)$. Therefore, $\text{val}(f^*) = f^*_{\text{out}}(s) = g^*(t_{\text{out}}, s_{\text{in}})$. Because this edge has a capacity of $\lceil \text{val}(f) \rceil$, we have $\text{val}(f^*) \leq \lceil \text{val}(f) \rceil$.

Finally, we analyze the time complexity. The auxiliary graph G' has $2|V| + 2$ vertices and $|E| + 3|V| + 1$ edges. Running the Ford–Fulkerson algorithm on G' to find g^* takes $O(|E'| \cdot \text{val}(g^*)) = O((|E| + |V|) \sum_{v \in V} L(v))$ time. Translating g^* back into f^* takes $O(|E|)$ time. This gives the claimed overall time complexity.

D Proof of Theorem 3

We begin by setting up the necessary preliminaries for the general scenario of arbitrary $k \geq 2$ and $n \geq k + 2$. We first generalize our graph formulation of the problem from the case $k = 2$. For each pair of candidates $c, c' \in C$, let $t_{c,c'}$ be the maximum integer such that c is ranked higher than c' in at least $t_{c,c'} \cdot n/k$ input rankings. We write $c \xrightarrow{t_{c,c'}} c'$ to mean that there is a constraint that c must be ranked higher than c' in at least $t_{c,c'}$ of the k output rankings. If $t_{c,c'} = k$, we call such a constraint a *strong constraint*.

We model the constraints as a weighted directed graph whose vertex set is $C = [m]$. For each pair $c, c' \in C$, we add an edge $c \to c'$ with weight $t_{c,c'}$. We will refer to the edges in the graph and the constraints they are induced by interchangeably. We next define a family of graphs, which generalize the graph that we used for the case $(n, k) = (4, 2)$.

Definition 1. For any integers $r, s \geq 2$, and any $b \in [k-1]$ and $d \in [s-1]$, the graph $G(r, s, b, d)$ is defined as a weighted directed graph with vertex set $C = [2rs]$. This graph is composed of s cycles D_j for $j \in [s-1]_0$, where D_j consists of the vertices $u \in \{2rj + 1, 2rj + 2, \ldots, 2rj + 2r\}$ and, for each $j' \in [r]$, the edges

$$2rj + 2j' - 1 \xrightarrow{b} 2rj + 2j' \quad \text{and} \quad 2rj + 2j' \xrightarrow{k} 2rj + ((2j' + 1) \bmod 2r).$$

For each $j \in [s-1]_0$, let E_j and O_j be the set of even-numbered and odd-numbered vertices in D_j, respectively. Taking indices modulo s, for each $j \in [s-1]_0$ and $d' \in [d]$, the graph also has the edge $u \xrightarrow{k} v$ for all $u \in E_j$ and $v \in O_{j+d'}$.

With this notation, the graph that we used earlier for the case $(n, k) = (4, 2)$ is $G(3, 4, 1, 1)$.

Given $G(r, s, b, d)$, we say that a ranking π (of the candidates in C) *touches* a cycle D_j if it satisfies a constraint of the type $u \xrightarrow{b} v$ in D_j. We will prove a useful lemma regarding these graphs.

Lemma 4. *For any integers $r, s \geq 2$, and any $b \in [k-1]$ and $d \in [s-1]$, any ranking π that satisfies all strong constraints in $G(r, s, b, d)$ touches D_j for at most $s - d$ indices $j \in [s-1]_0$.*

Proof. Suppose for contradiction that π touches (at least) $s-d+1$ of the cycles D_j. Let these cycles be $D_{j_0}, \dots, D_{j_{s-d}}$, where $j_0 < \cdots < j_{s-d}$. For each $q \in [s-d]_0$, let $u_q \xrightarrow{b} v_q$ be a constraint in D_{j_q} that π satisfies; in particular, u_q is odd and v_q is even. Taking indices modulo $s-d+1$, we will show that for every $q \in [s-d]_0$, there exists $d' \in [d]$ such that $j_{q+1} \equiv j_q + d' \pmod{s}$. Since $j_0 < \cdots < j_{s-d}$, there exists $d'' \in [s-d, s-1]$ such that $j_q \equiv j_{q+1} + d'' \pmod{s}$. Therefore, $j_{q+1} \equiv j_q + s - d'' \pmod{s}$, and we can take $d' = s - d''$.

By definition of $G(r, s, b, d)$, this implies that we have a cycle

$$u_0 \xrightarrow{b} v_0 \xrightarrow{k} u_1 \xrightarrow{b} v_1 \xrightarrow{k} \cdots \xrightarrow{k} u_{s-d} \xrightarrow{b} v_{s-d} \xrightarrow{k} u_0$$

in which every constraint is satisfied by π, which is a contradiction. $\square$

We now proceed to prove Theorem 3 in its full generality.

Proof of Theorem 3. Fix $k \geq 2$ and $n \geq k+2$. If $\gcd(k, n) \neq 1$, let $k' = \frac{k}{\gcd(k,n)} \in [k-1]$, so $k'n \equiv 0 \pmod{k}$. Else, $\gcd(k, n) = 1$; in this case, let k' be the inverse of $-n$ modulo k, so $k' \in [k-1]$ and $k'n \equiv -1 \pmod{k}$. In either case, let $n' = \left\lceil \frac{k'n}{k} \right\rceil$, which is the number of input rankings in which a candidate c must be ranked higher than another candidate c' in order to have a constraint $c \xrightarrow{k'} c'$. We have $n' \leq \frac{k'n}{k} + \frac{1}{k}$, which means that $kn' \leq k'n + 1$. Hence, $kn' + k \leq k'n + k + 1 < k'n + n$, so $k' + 1 > \frac{k(n'+1)}{n}$ and $\left\lfloor \frac{k(n'+1)}{n} \right\rfloor \leq k'$. Since $k' < k$, this implies that $n' + 1 < n$, and thus $n - n' - 1 \in [n-1]$. Therefore, we can consider the graph $G(n'+1, n, k', n-n'-1)$. This graph has $m = 2n(n'+1)$ vertices, with each vertex corresponding to a candidate.

We first show that the constraints in $G(n'+1, n, k', n-n'-1)$ cannot be satisfied by any k output rankings. For the sake of contradiction, assume otherwise. By Lemma 4, each output ranking can touch at most $n - (n - n' - 1) = n' + 1$ cycles in the graph. By the pigeonhole principle, there is a cycle touched by at most $\left\lfloor \frac{k(n'+1)}{n} \right\rfloor \leq k'$ rankings. Let D_j be a cycle touched by at most k' output rankings. If it is touched by at least one output ranking, consider one of these rankings π. The ranking π cannot satisfy every constraint in the cycle simultaneously, so there is a constraint $u \xrightarrow{k'} v$ in the cycle which is not satisfied by π. Note that only rankings that touch D_j can satisfy this constraint, so at most $k'-1$ output rankings satisfy the constraint. Else, if D_j is not touched by any output ranking, then since $k' - 1 \geq 0$, any constraint $u \xrightarrow{k'} v$ in D_j is again satisfied by at most $k'-1$ rankings. In either case, there exists a constraint $u \xrightarrow{k'} v$ in D_j that is not satisfied by the k output rankings.

It remains to construct n input rankings that generate the constraint graph $G(n'+1, n, k', n-n'-1)$. For each $j \in [n-1]_0$ and $j' \in [2n'+2]$, let $S_{j,j'}$ be the sequence of vertices that we obtain by traversing the cycle D_j starting from vertex $(2n'+2)j+j'$ and following its edges. Taking indices modulo n, for each $j \in [n-1]_0$, we define[4]

$$\begin{aligned}\pi_j = (&E_j, E_{j+1}, \dots, E_{j+n-n'-2},\\ &S_{j+n-n'-1,2}, S_{j+n-n',4}, \dots, S_{j+n-1,2n'+2},\\ &O_j, O_{j+1}, \dots, O_{j+n-n'-2}),\end{aligned}$$

where the elements in each $E_{j''}$ and $O_{j''}$ are arbitrarily ordered. We take $\pi_0, \pi_1, \dots, \pi_{n-1}$ as our n input rankings. Since $n'+2 \le n$, there is at least one E_j and at least one O_j in the ranking.

We first verify that for each $j \in [n-1]_0$, the ranking π_j satisfies all strong constraints.

- For all $j' \in [n-1]_0$ and $a \in [n'+1]$, the sequence $S_{j',2a}$ satisfies all strong constraints in $D_{j'}$. For each $j' \in [n-1]_0$, either $E_{j'}$ is ranked higher than $O_{j'}$ in π_j, or π_j contains $S_{j',2a}$ for some $a \in [n'+1]$, so all strong constraints in $D_{j'}$ are satisfied.
- For all[5] $j' \in [j, j+n-n'-2]$ and $d' \in [n-n'-1]$, we have that $E_{j'}$ is ranked higher than either $S_{j'+d',2a}$ (for some $a \in [n'+1]$) or $O_{j'+d'}$. Similarly, for all $j' \in [j+n-n'-1, j+n-1]$ and $d' \in [n-n'-1]$, we have that $S_{j',2a}$ (for some $a \in [n'+1]$) is ranked higher than either $S_{j'+d',2a'}$ (for some $a' \in [n'+1]$) or $O_{j'+d'}$. Therefore, the strong constraints across cycles, which are those from E_j to $O_{j+d'}$ for $j \in [n-1]_0$ and $d' \in [n-n'-1]$, are also satisfied.

Finally, observe that for each $j \in [n-1]_0$, the $n'+1$ sequences $S_{j,2}, S_{j,4}, \dots, S_{j,2n'+2}$ collectively satisfy each constraint of the type $u \xrightarrow{k'} v$ in D_j exactly n' times, and they appear in $\pi_{j-(n-n'-1)}, \dots, \pi_{j-(n-1)}$, respectively. Hence, all constraints $u \xrightarrow{k'} v$ in the graph are also satisfied. It follows that these n input rankings generate all constraints in the graph, as desired. $\square$

E Proofs of Theorems 4 and 5

For ease of reference, we provide the definitions of the two restricted classes here, and refer to the survey by Elkind et al. [18] for further discussion.

Definition 2 (Single-peaked)**.** A ranking π of the candidates in C is said to be *single-peaked with respect to* $(c_1, \dots, c_m)$, where $(c_1, \dots, c_m)$ is an ordering of the candidates in C, if there exists $\ell \in [m]$ such that c_ℓ is ranked top in π, and whenever $1 \le \ell'' < \ell' < \ell$ or $\ell < \ell' < \ell'' \le m$, the candidate $c_{\ell'}$ is ranked higher than $c_{\ell''}$ in π. The candidate c_ℓ is called the *peak* of π (or of a voter with the ranking π).

A set of rankings is said to be *single-peaked* if there exists an ordering $(c_1, \dots, c_m)$ of the candidates in C such that each ranking is single-peaked with respect to this ordering. An instance is called *single-peaked* if its set of input rankings is single-peaked.

Definition 3 (Single-crossing)**.** A set of rankings is said to be *single-crossing with respect to* $(i_1, \dots, i_n)$, where $(i_1, \dots, i_n)$ is an ordering of the voters in N, if for each pair of distinct candidates c and c', the voters who prefer c to c' form an interval in the ordering. A set of rankings is said to be *single-crossing* if it is single-crossing with respect to some ordering of the voters in N. An instance is called *single-crossing* if its set of input rankings is single-crossing.

Before starting the proof of Theorem 4 for single-peaked preferences, we will set up some preliminaries. Without loss of generality, assume that the set of input rankings is single-peaked with respect to the ordering $(1, 2, \dots, m)$. We start with two simple but useful lemmas.

Lemma 5. *For candidates $c < c' < c''$, an input ranking that ranks c higher than c' also ranks c' higher than c'' (and therefore ranks c higher than c'' as well). Similarly, an input ranking that ranks c'' higher than c' also ranks c' higher than c (and therefore ranks c'' higher than c as well).*

Proof. Since c is ranked higher than c', the peak of the ranking must be some candidate $a < c'$. Then, $a < c' < c''$, and so c' is ranked higher than c''. The other claim can be shown analogously. $\square$

[4] This is slightly different from the definition that we used for the case $(n, k) = (4, 2)$.

[5] Again, these indices are taken modulo n.

Algorithm 1 Topological Sort of G_d

Input: Directed graph G_d with vertex set $C = [m]$.
Output: Topological ordering of G_d which is single-peaked with respect to the ordering $(1, 2, \dots, m)$.
$a \leftarrow 1$, $b \leftarrow m$
$\pi_d \leftarrow (\ )$
while $a \leq b$ **do**
  if there does not exist a candidate $c \in [a+1, b]$ such that a has an outgoing edge to c **then**
    Prepend a to the front of π_d.
    $a \leftarrow a + 1$
  else
    Prepend b to the front of π_d.
    $b \leftarrow b - 1$
return π_d

Lemma 6. *For candidates $c < c' < c''$, an input ranking that ranks c higher than c'' also ranks c' higher than c''. Similarly, an input ranking that ranks c'' higher than c also ranks c' higher than c.*

Proof. Given that c is ranked higher than c'', the peak must be a candidate $a < c''$. If $c' < a < c''$, then c' is ranked higher than c, and thus c' is ranked higher than c''. If $a \leq c'$, then c' is always ranked higher than c''. The other claim can be proven analogously. □

We will use the same notation for constraints as in the proof of the general case of Theorem 3 (see Appendix D). In particular, for candidates $c, c' \in C$, an edge $c \xrightarrow{t} c'$ for some $t \in [k]_0$ means that t is the largest integer such that c is ranked higher than c' in at least $t \cdot n/k$ input rankings, and consequently c must be ranked higher than c' in at least t output rankings.

For each $d \in [k]$, let G_d be an unweighted directed graph with vertex set $C = [m]$, such that for each pair of candidates $c < c'$ we have an edge $c \rightarrow c'$ if and only if there is a constraint $c \xrightarrow{t} c'$ for some $t \in [d, k]$, and for each pair of candidates $c > c'$ we have an edge $c \rightarrow c'$ if and only if there is a constraint $c \xrightarrow{t} c'$ for some $t \in [k - d + 1, k]$. For each $d \in [k]$, perform Algorithm 1 on G_d, which outputs a ranking π_d.

We prove the following proposition about π_d.

Proposition 2. *For each $d \in [k]$, the ranking π_d is a topological ordering of G_d and is single-peaked with respect to the ordering $(1, 2, \dots, m)$.*

Proof. Firstly, note that at the start of each iteration, $[a, b]$ represents the set of candidates that have not yet been added to π_d. Since we always add either a or b to the front of π_d and recurse to either $[a+1, b]$ or $[a, b-1]$, respectively, the final ranking π_d is guaranteed to be single-peaked with respect to the ordering $(1, 2, \dots, m)$.

It remains to show that π_d is a topological ordering of G_d. Throughout Algorithm 1, we show that the following invariants always hold:

(1) No candidate already added to π_d has an outgoing edge to any candidate in $[a, b]$.

(2) For candidates c, c' such that c is (not necessarily directly) in front of c' in π_d, it holds that c' does not have an outgoing edge to c.

If these invariants are maintained throughout the algorithm, then invariant (2) implies that π_d is a topological ordering of G_d, as π_d contains all candidates at the end of the algorithm. The invariants trivially hold at the start of the algorithm since π_d is empty. Consider the two cases for the conditional branch in the algorithm.

Case 1: a has no outgoing edge to any candidate in $[a+1, b]$.

Invariant (1) implies that if we add a to the front of π_d, invariant (2) is still preserved. Also, by the premise of this case, invariant (1) is still preserved after adding a.

Case 2: a has an outgoing edge to some candidate $c \in [a+1, b]$.

Since we now add b to the front of π_d instead of a, it suffices to show that b has no outgoing edge to any candidate in $[a, b-1]$, after which the argument follows as in Case 1.

Suppose for contradiction that b has an outgoing edge to some candidate $c' \in [a, b-1]$. Since $c' < b$, by definition of G_d, this outgoing edge corresponds to a constraint $b \xrightarrow{t} c'$ where $k - d + 1 \leq t \leq k$. This means that at least $\frac{tn}{k} \geq \frac{(k-d+1)n}{k}$ input rankings rank b higher than c'. As $a \leq c' < b$, by Lemma 5, at least $\frac{(k-d+1)n}{k}$ input rankings rank b higher than a. Similarly, by the premise of this case, we can deduce that at least $\frac{dn}{k}$ input rankings rank a higher than b.

However, an input ranking cannot rank a higher than b and b higher than a simultaneously, so there are at least $\frac{(k-d+1)n}{k} + \frac{dn}{k} = \frac{(k+1)n}{k} > n$ input rankings, a contradiction. It follows that both invariants are maintained at every step, and π_d is indeed a topological ordering of G_d. □

We next present a lemma related to positional proportionality.

Lemma 7. *Let $c \in C$ be a candidate and $d \in [k]_0$. Let $\ell \in [m]$ and $t \in [k]$ be such that c appears in the top ℓ ranks in at least $t \cdot n/k$ input rankings. Let x_d be the number of rankings among $\pi_1, \ldots, \pi_d$ (which is an empty sequence if $d = 0$) where c appears in the top ℓ ranks. Then, at least one of the following holds:*

(1) $x_d \geq t$.

(2) There are at least $\frac{(d-x_d+t)n}{k}$ input rankings such that each of these rankings either ranks at least one candidate in[6] $\{1, 2, \ldots, c-\ell\}$ higher than c or ranks c in the top ℓ ranks. (It is possible that $c - \ell \leq 0$, in which case the set is empty.)

To prove this lemma, we first need to establish another lemma.

Lemma 8. *Let $c \in C$ be a candidate and $\ell \in [m]$. Consider an input ranking that ranks at least one candidate in $\{1, 2, \ldots, c-\ell\}$ higher than c or ranks c in the top ℓ ranks. Then, this ranking also ranks c higher than every candidate in $\{c+\ell, c+\ell+1, \ldots, m\}$.*

Proof. By the single-peaked property, the candidates in the top ℓ ranks in each ranking form a contiguous range $[a, a+\ell-1]$ for some $a \in [m-\ell+1]$. For rankings in which c appears in the top ℓ ranks, we must have $a \leq c$, and so $a + \ell - 1 < c + \ell$. Hence, these rankings rank c higher than all candidates in $\{c+\ell, \ldots, m\}$, as all such candidates appear outside the top ℓ ranks. On the other hand, by Lemma 5, rankings that rank at least one candidate in $\{1, \ldots, c-\ell\}$ higher than c also rank c higher than all candidates in $\{c+\ell, \ldots, m\}$. □

With Lemma 8 in hand, we can now prove Lemma 7.

Proof of Lemma 7. We proceed by induction on d. For the base case $d = 0$, we have $x_0 = 0$, and so $d - x_d + t = t$. Since c appears in the top ℓ ranks in at least $t \cdot n/k$ input rankings, the lemma holds for the base case. Assume that the inductive hypothesis is true up to some $d \in [k-1]_0$. If $x_d \geq t$, then we also have $x_{d+1} \geq t$, which completes the inductive step for this case.

Suppose from now on that $x_d < t$, so property (2) must be true for d by the inductive hypothesis. By definition, we have $x_{d+1} \in \{x_d, x_d + 1\}$. If $x_{d+1} = x_d + 1$, then $d + 1 - x_{d+1} + t = d - x_d + t$, so property (2) is still true for $d + 1$. Otherwise, $x_{d+1} = x_d$, and so c is not in the top ℓ ranks of π_{d+1}. Suppose that in the iteration of Algorithm 1 when c is added to π_{d+1}, the unranked candidates form a range $[a, b]$. Since c is added at this step, we have $c = a$ or $c = b$.

Case 1: $c = a$.

As c is not in the top ℓ ranks of π_{d+1}, we have $b \geq c+\ell$. Since $x_d < t$, it holds that $d - x_d + t \geq d+1$, so by Lemma 8, for each $c'' \in \{c+\ell, \ldots, m\}$, there exists $t' \in [d+1, k]$ such that there is a constraint

[6] Since $c - \ell$ may be negative, in which case the set is empty, we avoid using the notation $[c - \ell]$ here.

$c \xrightarrow{t'} c''$. By definition of G_{d+1}, there is an edge $c \to c''$ for each $c'' \in \{c+\ell, \dots, m\}$. As c cannot have an outgoing edge to any candidate in $[c+1, b]$ when we add c to the topological ordering, it holds that $b \leq c + \ell - 1$, which contradicts $b \geq c + \ell$. Hence, this case cannot occur.

Case 2: $c = b$.

As c is not in the top ℓ ranks of π_{d+1}, we have $a \leq c - \ell$. By our algorithm, we only choose b as the next candidate to be added if a has an outgoing edge to some candidate in $[a+1, b]$. Thus, by definition of G_{d+1}, there is a constraint $a \xrightarrow{t'} c'$ for some $c' \in [a+1, b]$ and $t' \in [d+1, k]$. By Lemma 5, all input rankings that rank a higher than c' also rank a higher than $b = c$, so at least $\frac{(d+1)n}{k}$ input rankings rank a higher than c.

Note that input rankings that rank c in the top ℓ positions cannot rank a higher than c. Indeed, the top ℓ positions must form a contiguous range of ℓ candidates containing c, and a is outside this range since $a \leq c - \ell$. Thus, there are at least $\frac{(d+1)n}{k} + \frac{tn}{k} = \frac{(d+1+t)n}{k}$ input rankings that rank a higher than c or rank c in the top ℓ positions. Since $d + 1 + t \geq d + 1 - x_{d+1} + t$, property (2) still holds for $d + 1$. This completes the induction. □

Next, we introduce a lemma related to PSC, which is analogous to Lemma 7 for positional proportionality.

Lemma 9. *Let $C' = [a, b]$ for some $a, b \in C$ with $a \leq b$, and $d \in [k]_0$. Let $t \in [k]$ be such that all candidates in C' appear in the top $|C'|$ ranks in at least $t \cdot n/k$ input rankings. Let x_d be the number of rankings among $\pi_1, \dots, \pi_d$ (which is an empty sequence if $d = 0$) where all candidates in C' appear in the top $|C'|$ ranks. Then, at least one of the following holds:*

(1) $x_d \geq t$.

(2) There are at least $\frac{(d - x_d + t)n}{k}$ input rankings such that each of these rankings either ranks at least one candidate in $\{1, 2, \dots, a-1\}$ higher than b or ranks all candidates in C' in the top $|C'|$ ranks.

To prove this lemma, we again need to establish another lemma.

Lemma 10. *Let $C' = [a, b]$ for some $a, b \in C$ with $a \leq b$. Consider an input ranking that ranks at least one candidate in $\{1, 2, \dots, a-1\}$ higher than b or ranks all candidates in C' in the top $|C'|$ ranks. Then, this ranking also ranks a higher than all candidates in $\{b+1, \dots, m\}$.*

Proof. If an input ranking ranks all candidates in C' in the top $|C'|$ ranks, then all other candidates are outside the top $|C'|$ ranks, and so all candidates in C' (including a) are ranked higher than all candidates in $\{b+1, \dots, m\}$.

On the other hand, if an input ranking ranks some candidate $c \in \{1, 2, \dots, a-1\}$ higher than b, by Lemma 5, it ranks b higher than all candidates in $\{b+1, \dots, m\}$. Since $c < b$ and c is ranked higher than b, by Lemma 6, all candidates in $C' \setminus \{b\}$ are ranked higher than b. It follows that $a \in C'$ is ranked higher than all candidates in $\{b+1, \dots, m\}$. □

With Lemma 10 in hand, we can now prove Lemma 9.

Proof. We proceed by induction on d. For the base case $d = 0$, we have $x_0 = 0$, and so $d - x_d + t = t$. Since all candidates C' appear in the top $|C'|$ ranks in at least $t \cdot n/k$ input rankings, the lemma holds for the base case. Assume that the inductive hypothesis is true up to some $d \in [k-1]_0$. If $x_d \geq t$, then we also have $x_{d+1} \geq t$, which completes the inductive step for this case.

Suppose from now on that $x_d < t$, so property (2) must be true for d by the inductive hypothesis. By definition, we have $x_{d+1} \in \{x_d, x_d + 1\}$. If $x_{d+1} = x_d + 1$, then $d + 1 - x_{d+1} + t = d - x_d + t$, so property (2) is still true for $d + 1$. Otherwise, $x_{d+1} = x_d$, and so not all candidates in C' appear in the top $|C'|$ ranks of π_{d+1}. Observe that by Algorithm 1, the candidates in the top $|C'|$ ranks of π_{d+1} must form an interval $[a', b']$ such that $a' \neq a$ and $b' - a' = b - a$.

Case 1: $a' > a$.

Since $b' - a' = b - a$, we have $b' > b$. When a is added to π_{d+1}, the unranked candidates must form an interval $[a, b'']$ for some $b'' \in [b', m]$. This means that there is no outgoing edge from a to any candidate in $[a+1, b'']$.

Since $x_d < t$, it holds that $d - x_d + t \geq d + 1$, so by Lemma 10, for each $c \in \{b+1, \ldots, m\}$, there exists $t' \in [d+1, k]$ such that there is a constraint $a \xrightarrow{t'} c$. However, $b'' \in \{b+1, \ldots, m\}$, so by definition of G_{d+1}, there is an outgoing edge from a to b'', a contradiction. Hence, this case cannot occur.

Case 2: $a' < a$.

Since $b' - a' = b - a$, we have $b' < b$. When b is added to π_{d+1}, the unranked candidates must form an interval $[a'', b]$ for some $a'' \in [a']$. Since b is added at this step, a'' must have an outgoing edge to some candidate in $[a''+1, b]$. By definition of G_{d+1}, there is a constraint $a'' \xrightarrow{t'} c$ for some $c \in [a''+1, b]$ and $t' \in [d+1, k]$, which implies that at least $\frac{(d+1)n}{k}$ input rankings rank a'' higher than c. By Lemma 5, at least $\frac{(d+1)n}{k}$ input rankings rank a'' higher than b.

Note that input rankings that rank a'' higher than b cannot rank all candidates in C' in the top $|C'|$ ranks, since $a'' \notin C'$ and $b \in C'$. Thus, there are at least $\frac{(d+1)n}{k} + \frac{tn}{k} = \frac{(d+1+t)n}{k}$ input rankings that rank a'' higher than b or rank all candidates in C' in the top $|C'|$ ranks. Since $d + 1 + t \geq d + 1 - x_{d+1} + t$, property (2) still holds for $d + 1$. This completes the induction. □

Finally, we are ready to prove Theorem 4.

Proof of Theorem 4. Let the k output rankings be $\pi_1, \pi_2, \ldots, \pi_k$. As shown in Proposition 2, these rankings are single-peaked with respect to the ordering $(1, 2, \ldots, m)$.

First, we verify that this output satisfies pairwise proportionality. If we have a constraint $c \xrightarrow{t} c'$ for some $c < c'$ and $t \in [k]$, then the graphs $G_1, \ldots, G_t$ contain the edge $c \to c'$, and thus c is ranked higher than c' in the t rankings $\pi_1, \ldots, \pi_t$ since they are topological orderings of $G_1, \ldots, G_t$, respectively. Similarly, if we have a constraint $c \xrightarrow{t} c'$ for some $c > c'$ and $t \in [k]$, then the graphs $G_{k-t+1}, \ldots, G_k$ contain the edge $c \to c'$, and thus c is ranked higher than c' in the t rankings $\pi_{k-t+1}, \ldots, \pi_k$.

Next, we verify that this output satisfies positional proportionality. Suppose that c appears in the top ℓ ranks in at least $t \cdot n/k$ input rankings, for some $\ell \in [m]$ and $t \in [k]$. By Lemma 7, using the same definition of x_k, if $x_k < t$, then there are at least $\frac{(k - x_k + t)n}{k} \geq \frac{(k+1)n}{k} > n$ input rankings, which is clearly false. Hence, $x_k \geq t$, which precisely means that c appears in the top ℓ ranks in at least t of $\pi_1, \ldots, \pi_k$.

We can show that this output also satisfies PSC by using Lemma 9 analogously. Indeed, observe that for each input ranking and any $\ell \in [m]$, the candidates in the top ℓ positions form an interval $C' = [a, b]$ for some $a, b \in C$ with $b = a + \ell - 1$, due to the single-peaked property. Hence, Lemma 9 can be applied to any solid coalition of voters.

It remains to show that we can construct $\pi_1, \ldots, \pi_k$ in polynomial time. For each pair of candidates c and c', let $t_{c,c'}$ be the maximum integer such that c is ranked higher than c' in at least $t_{c,c'} \cdot n/k$ input rankings. We can compute $t_{c,c'}$ for all $c, c' \in C$ in $O(nm^2)$ time, which allows us to construct the adjacency matrices of the graphs $G_1, \ldots, G_k$ in $O(m^2 k)$ time. Each run of Algorithm 1 takes $O(m^2)$ time, and since we run it for all $d \in [k]$, the runs take $O(m^2 k)$ time in total. It follows that we can output $\pi_1, \ldots, \pi_k$ in $O(m^2(n + k))$ time. □

Next, we prove Theorem 5 for single-crossing preferences. We need the following technical lemma.

Lemma 11. *Let $a, b \in [n]$ be such that $b \geq a + \lceil t \cdot n/k \rceil - 1$ for some $t \in [k]$. Then, the interval $[a, b]$ contains $\lceil d \cdot n/k \rceil$ for at least t values of $d \in [k]$.*

Proof. Observe that for each $r \in [n]$, the set $[r]$ contains $\lceil d \cdot n/k \rceil$ for exactly $\lfloor rk/n \rfloor$ values of $d \in [k]$. Therefore, the interval $[a, b]$ contains $\lceil d \cdot n/k \rceil$ for $\lfloor bk/n \rfloor - \lfloor (a-1)k/n \rfloor \geq$

$\lfloor (b-a+1)k/n \rfloor \geq t$ values of $d \in [k]$, where we use the fact that $\lfloor y+z \rfloor \geq \lfloor y \rfloor + \lfloor z \rfloor$ for any real numbers y, z. ☐

We proceed to the proof of Theorem 5.

Proof of Theorem 5. Without loss of generality, assume that the set of input rankings is single-crossing with respect to the ordering $(1, 2, \ldots, n)$. For each $d \in [k]$, let σ_d be the ranking $\pi_{\lceil d \cdot n/k \rceil}$ submitted by voter $\lceil d \cdot n/k \rceil$. We let $\sigma_1, \ldots, \sigma_k$ be the k output rankings.

From the single-crossing definition, for any pair of candidates $c, c' \in C$, the set of voters $N' \subseteq N$ who prefer c to c' forms either a prefix or a suffix of $(1, 2, \ldots, n)$. Similarly, for any subset of candidates $C' \subseteq C$, the set of voters who rank the candidates from C' in the top $|C'|$ ranks forms an interval of $(1, 2, \ldots, n)$ (but not necessarily a prefix or suffix). Indeed, if two voters rank the candidates from C' in the top $|C'|$ ranks, then so does every voter between them.

We will establish pairwise proportionality; the proof for PSC is analogous. Suppose that for some $t \in [k]$, at least $t \cdot n/k$ voters prefer candidate c to candidate c'. These voters must form an interval $[a, b]$ of $(1, 2, \ldots, n)$, where $b \geq a + \lceil t \cdot n/k \rceil - 1$. By Lemma 11, the rankings of at least t of these voters are chosen in the output, which means that pairwise proportionality is satisfied.

Finally, we can trivially output the rankings in $O(mk)$ time by computing $\lceil d \cdot n/k \rceil$ for each $d \in [k]$ and outputting the corresponding input ranking. ☐

It is worth noting that the other positive results in this paper *cannot* be achieved by selecting the output solely from the input rankings. Moreover, for single-crossing instances, this output restriction also renders positional proportionality unattainable. We refer to Appendices A and F for detailed discussions.

F Choosing output from input rankings for single-crossing instances

For single-crossing instances, one may hope that the approach used in Theorem 5 to achieve pairwise proportionality and PSC could also be used to achieve positional proportionality. We show not only that this approach fails, but also that it is fundamentally impossible to attain positional proportionality for single-crossing instances by choosing from the input. Consider the instance below with $(n, m, k) = (4, 14, 2)$.

$$\pi_1 = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14);$$
$$\pi_2 = (1, 3, 2, 4, 5, 7, 6, 8, 10, 9, 11, 12, 14, 13);$$
$$\pi_3 = (3, 2, 1, 4, 5, 8, 7, 6, 10, 9, 11, 12, 14, 13);$$
$$\pi_4 = (3, 4, 2, 1, 8, 7, 5, 6, 11, 10, 9, 14, 13, 12).$$

Intuitively, this instance is composed of four non-crossing sets of candidates: $\{1, 2, 3, 4\}$, $\{5, 6, 7, 8\}$, $\{9, 10, 11\}$, $\{12, 13, 14\}$. One can verify that the instance is indeed single-crossing by checking the pairwise relations within each set of candidates. Now, for each pair of input rankings, we show that there exist $c \in C$ and $\ell \in [m]$ such that candidate c appears in the top ℓ ranks *only* within that pair.

- π_1, π_2: candidate 1 appears in the top 1 rank.
- π_3, π_4: candidate 3 appears in the top 1 rank.
- π_1, π_3: candidate 2 appears in the top 2 ranks.
- π_2, π_4: candidate 7 appears in the top 6 ranks.
- π_1, π_4: candidate 13 appears in the top 13 ranks.
- π_2, π_3: candidate 10 appears in the top 9 ranks.

Each pair imposes a positional constraint on the output that requires selecting at least one of those two rankings. Because no output of size two can intersect all six pairs, positional proportionality cannot be achieved.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction reflect the results presented in this paper.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of the work presented in this paper are discussed as future directions in Section 6.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: For each theoretical result, we provide a formal statement as well as a complete proof.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [N/A]

   Justification: The paper does not include experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [N/A]

Justification: The paper does not include experiments.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [N/A]

Justification: The paper does not include experiments.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [N/A]

Justification: The paper does not include experiments.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [N/A]

   Justification: The paper does not include experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in this paper is entirely theoretical and does not involve human subjects or data.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This paper presents foundational research and is not tied to particular applications, let alone deployments. We are not aware of a direct path to any negative applications.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not involve data or models.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [N/A]

    Justification: The paper does not use existing assets.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.