SIMULATION-AUGMENTED GENERATION VIA REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Simulation-augmented generation (Milli et al., 2026) is a recent framework for answering normative questions by querying generative simulations of individuals in a prompt-dependent target population and synthesizing their responses. Kraiczy et al. (2026) formalize representative selection in this setting using social choice and, to avoid simulating every individual, use proportional clustering over predicted viewpoint embeddings to route each prompt to a small subset k among n simulations. We introduce reinforcement learning from social choice axioms (RL-SCA), a framework for directly training representative routers using continuous measures of social choice guarantees as reward signals. In our setting, we use the mEJR+ factor, a continuous measure of proportional representation, and show that it can be computed in $O(n^2(\log n + k))$ time, making it practical as a reward for reinforcement learning. This allows us to train a lightweight router end-to-end to map each prompt directly to a representative subset of simulations, replacing the two-stage process of predicting population-wide proxy embeddings and subsequently clustering them. The resulting router requires only a single forward pass at inference time, avoiding proxy-embedding computations for the full population while achieving significantly improved proportional representation across three datasets spanning politics, personal advice and book reviews.

1 INTRODUCTION

To give users greater control and transparency over whose viewpoints are represented in model responses to questions without a single ground-truth answer, Milli et al. (2026) recently introduced *simulation-augmented generation* (SAGE). SAGE augments generation by querying simulations of individuals from an explicit, prompt-dependent target population and incorporating their perspectives into the final response. For example, given the prompt "Should schools use standardized testing?", a model might identify an appropriate population, such as the general public or domain experts, and ask simulated individuals what considerations matter most. Their responses might highlight concerns such as teaching to the test or effects on student motivation. The model then synthesizes these perspectives into a response that reflects the range of viewpoints present in the target population.

The routing challenge and formalization. Making SAGE practical for inference-time use requires keeping simulation latency and compute costs low. A natural approach is *routing*: instead of querying all n simulations, a router uses the prompt to dynamically select a small representative subset of $k \ll n$ individuals to simulate, producing a *slate* of k responses. The challenge is ensuring this slate genuinely represents what all n would have produced in a principled way. To formalize this, Kraiczy et al. (2026) utilize the notion of a *viewpoint embedding space*, where distances reflect human preference judgments, rather than mere semantic similarity (e.g., as recently explored by Blair et al. (2026)). Once responses are mapped to this embedding space, whether a subset of k responses is representative can be assessed using existing social choice axioms for proportional clustering. Specifically, they utilize metric proportional justified representation+ (mPJR+) (He et al., 2026; Kellerhals & Peters, 2024), which is the strongest proportionality axiom currently known to always be satisfiable by centroid-based clustering.

Prior work and its limitations. Conceptually, the router of Kraiczy et al. (2026) is a direct adaptation of existing work on proportional clustering. Their method uses a variant of SEAR (Aziz et al.,

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

2024), which is guaranteed to satisfy mPJR+He et al. (2026) when applied to the ground-truth response embeddings. Because obtaining each ground-truth embedding requires querying a simulation and then embedding their response, Kraiczy et al. (2026) replace these embeddings with predicted embeddings for efficiency. Specifically, their router predicts n proxy viewpoint embeddings, one for each individual, runs the proportional clustering procedure on these predicted embeddings to select k representatives, and then queries only the k corresponding simulations to get their open-ended responses.

This approach has two limitations that we address in this paper. First, although it reduces the number of simulation queries from n to k, it still requires computing n predicted embeddings. Second, its mPJR+ satisfaction rate degrades substantially as k increases, falling to 40% at $k = 10$, a practically relevant value of k. One possible explanation is that, as k increases, smaller groups of size at least n/k become entitled to representation under mPJR+, and these smaller groups may be more sensitive to noise in the proxy embeddings. Because the appropriate value of k may vary across prompts and applications, representation quality should remain robust across a practically relevant range of k.

Our approach: learning a router from social choice axioms. We adopt the formalization of representative routing as selection or representative viewpoints in a viewpoint embedding space introduced by Kraiczy et al. (2026), but propose an alternative method that substantially improves on both computational efficiency *and* representativeness. As an analogy, consider mixture-of-experts (MoE) models: the router in MoE models is typically a single linear layer mapping the input token embedding to a distribution over experts, from which k are selected (Shazeer et al., 2017). Drawing inspiration from this paradigm, we learn a lightweight MLP with two hidden layers that maps a prompt embedding directly to scores over the n simulations, which induce a distribution according to which k are selected during RL training. At inference-time, we select the highest scoring k simulations, requiring only a single forward pass.

The central challenge is then how to train such a router to select proportionally representative slates. Doing so requires more than an axiom whose binary satisfaction can be checked. We need a proportionality objective that can be evaluated efficiently and that provides a graded measure of how well a slate satisfies proportionality. Exact mPJR+ is not practically verifiable at the scale required for repeated RL updates.[1] More generally, binary satisfaction of any proportionality axiom provides a coarse learning signal because all violating slates receive the same value, leaving sampled batches with little or no reward variation.

We address both issues using metric EJR+ (mEJR+) (Brill & Peters, 2023a; Kellerhals & Peters, 2024), a stronger proportionality axiom that implies mPJR+. We first show that mEJR+ admits efficient binary verification in $O(n^2(\log n + k))$ time. Moreover, we show that in the same running time we can compute the *mEJR+ factor*: the smallest multiplicative relaxation under which a given slate satisfies mEJR+. This turns a binary social-choice axiom into a continuous, verifiable measure of proportional representation. The factor distinguishes among violating slates according to how close they are to satisfying mEJR+, and remains informative even when exact mEJR+ is not satisfiable Brill & Peters (2023a).

We use the mEJR+ factor as a *verifiable reward* for reinforcement learning, yielding a paradigm we call ***reinforcement learning from social choice axioms*** **(RL-SCA)**. By optimizing proportional representation end-to-end, our router removes the intermediate proxy-embedding stage and directly optimizes representation while also eliminating the need to compute n such embeddings at inference time. We evaluate our router trained via RL-SCA across three domains—campus-protest opinions (*Remesh*, (Konya et al., 2023)), Reddit AITA discussions (*Humanual Opinion*, (Wu et al., 2026)), and book reviews (*Humanual Book*, (Wu et al., 2026)).

For our application, mEJR+ offers an attractive combination: it provides a strong proportionality guarantee while admitting the efficient quantitative verification needed for learning. We do not view mEJR+ as uniquely privileged—the same paradigm could in principle be instantiated with other proportionality axioms if similarly practical quantitative verifiers become available.

Our contributions are:

[1] Concurrent work by (He et al., 2026) gives a polynomial-time verification algorithm for mPJR+ based on submodular-function minimization; we discuss its computational implications in Section 2.

- **Efficient quantitative verification of proportionality.** We show that the mEJR+ factor—the smallest multiplicative relaxation under which a slate satisfies mEJR+—can be computed in $O(n^2(\log n + k))$ time. This also yields efficient verification of exact mEJR+ and makes mEJR+ well suited to settings requiring repeated evaluation.
- **Reinforcement learning from social choice axioms.** We introduce *reinforcement learning from social choice axioms* (RL-SCA), using the mEJR+ factor as a fine-grained, verifiable reward for proportional representation. To our knowledge, this is the first use of quantitative verification of a social-choice axiom as a reward signal for reinforcement learning.
- **End-to-end representative routing.** We train a lightweight router to select representatives directly from the prompt, eliminating the need to construct n intermediate proxy viewpoint embeddings. The router thereby optimizes proportional representation end-to-end while requiring only a single router forward pass at inference time.
- **Empirical evaluation.** We evaluate the RL-SCA across three datasets and against three baselines, finding a **19–29 percentage-point higher mPJR+ satisfaction rate** relative to the router of (Kraiczy et al., 2026).

1.1 Preliminaries

We begin by recalling the viewpoint-space routing framework of Kraiczy et al. (2026), which provides the starting point for our approach. In particular, we review their formulation of proportional representation in viewpoint space and their Predictive Route–then–Simulate procedure.

Proportional representation in viewpoint space. Let $N = [n]$ index a pool of simulations, and fix a slate size k with $1 \leq k < n$. A slate is a subset $X \subseteq N$ of size k. Our goal is to select a slate whose responses proportionally represent those of the full pool; following Kraiczy et al. (2026), we call this selection problem routing. Simulation A_i of human i on prompt p generates a response $A_i(p)$. A viewpoint embedding (VPE) function ψ maps responses into a metric space $(\mathcal{Z}, d)$ in which proximity reflects similarity of viewpoints rather than merely semantic similarity. We write $z_i(p) := \psi(A_i(p))$ for the viewpoint of simulation i on prompt p.

For a fixed prompt p, write $Z_p = (z_i(p))_{i \in N}$ for the generated viewpoint geometry. This induces a proportional-clustering instance in which the same indexed simulations serve as voters and candidate representatives. Distinct simulations count separately even when their embeddings coincide. Suppressing dependence on p, write $\Delta(i,j) := d(z_i, z_j)$. For each $\ell \in [k]$, let $q_\ell := \lceil \ell n/k \rceil$ be the group-size threshold for requiring ℓ representatives under the criterion below. Following Kraiczy et al. (2026), we use metric proportional justified representation plus (mPJR+) (He et al., 2026).

Definition 1 (Metric PJR+)**.** A slate X satisfies mPJR+ if, for every $\ell \in [k]$, every unselected candidate $c \in N \setminus X$, and every group $S \subseteq N$ with $|S| \geq q_\ell$,

$$\left| \left\{ x \in X : \min_{i \in S} \Delta(i,x) \leq \max_{i \in S} \Delta(i,c) \right\} \right| \geq \ell.$$

Fix $\ell \in [k]$, a group S with $|S| \geq q_\ell$, and an unselected candidate c. Let $r = \max_{i \in S} \Delta(i,c)$, so every member of S is within distance r of c. The mPJR+ condition requires at least ℓ selected representatives, each within distance r of some member of S. The member may differ between representatives. Thus, group size determines the number of representatives required, while distance to the unselected candidate determines the comparison radius. The rule SEAR (Aziz et al., 2024) always returns an mPJR+ slate.

Predictive Route–then–Simulate. If the generated viewpoint geometry Z_p were available, SEAR could select the k simulations directly. Obtaining Z_p, however, requires querying all n simulations and embedding their responses before selecting the slate.

To avoid this exhaustive generation step, Kraiczy et al. (2026, Sec. 4) introduce Predictive Route–then–Simulate. A predictor g_ω computes $\widehat{z}_i(p) := g_\omega(s_i, p)$ for every $i \in N$. SEAR then uses the predicted geometry $\widehat{Z}_p = (\widehat{z}_i(p))_{i \in N}$ to select k simulation indices, and only the selected simulations are queried. Selection therefore uses the predicted geometry $\widehat{Z}_p$, while proportional representation is assessed in the generated geometry Z_p. The procedure still requires predicting an embedding for every simulation before selection.

2 REINFORCEMENT LEARNING FROM SOCIAL CHOICE AXIOMS

Predictive Route–then–Simulate follows a predict–then–optimize decomposition: it first learns a predictor of simulation-response embeddings and then routes on the predicted geometry. The loss used to learn this intermediate representation, however, is not the quantity ultimately of interest—proportional representation of the selected slate. Prior work on decision-focused learning has shown that such a separation between prediction and downstream optimization can lead to suboptimal decisions, motivating end-to-end training on the downstream objective (Donti et al., 2017; Elmachtoub & Grigas, 2022; Cameron et al., 2022). We therefore collapse the two stages of PRS into a single learned router and train a policy to select representative slates directly.

Formally, we parameterize the router by a stochastic policy

$$\pi_\theta(X \mid p), \qquad X \in \binom{N}{k},$$

which induces a distribution directly over size-k slates. The learning objective is to place high probability on slates that are proportionally representative in the true viewpoint geometry. To obtain a reward for a sampled slate, we use the exhaustive viewpoint embeddings $Z_p = (z_i(p))_{i \in N}$, where $z_i(p) = \psi(A_i(p))$. These embeddings are needed only to *evaluate* the policy during training and can be obtained offline by querying the simulations and embedding their responses. Importantly, the exhaustive embeddings Z_p are used only to evaluate the policy during training and are not inputs to the router. At inference time, the router conditions only on the prompt p and selects the highest-probability slate under $\pi_\theta(\cdot \mid p)$, so no exhaustive simulation step is required. For a distribution $\mathcal{D}$ over prompts and a reward $R_p(X)$ computed from Z_p, we optimize $J(\theta) := \mathbb{E}_{p \sim \mathcal{D}} \mathbb{E}_{X \sim \pi_\theta(\cdot|p)} [R_p(X)]$. Thus, the policy maps the prompt p directly to a distribution over size-k slates of simulation indices.

This approach requires a proportionality objective that can be evaluated efficiently and repeatedly during training. We therefore turn to proportionality guarantees that admit sufficiently efficient verification to serve as training objectives.

Efficiently verifiable rewards for proportional representation. A natural approach would be to use mPJR+ satisfaction to obtain a reward $R_p(X)$ for reinforcement learning. However, existing verification algorithms are too expensive to use inside an RL training loop. In our setting, where the n simulation embeddings act as both voters and candidates, the polynomial-time verifier of He et al. (2026) has $O\Big(n^6 k \log^2 n + n^6 \log^{O(1)} n\Big)$[2] runtime, while their fixed-parameter algorithm runs in $O(n^2 \log n \, 2^k)$ time. Both are impractical to invoke for every sampled slate.[3]

We therefore consider metric EJR+ (mEJR+) (Kellerhals & Peters, 2024), the metric analogue of EJR+ from multi-winner voting (Brill & Peters, 2023b).Let $\Delta_\ell(i, X)$ denote the distance from i to its ℓth-closest member of X, and, for nonempty $S \subseteq N$, define $r_X(S) := \min_{c \in N \setminus X} \max_{i \in S} \Delta(i, c)$.

Definition 2 (Metric EJR+)**.** A slate X satisfies metric EJR+ (mEJR+) if, for every $\ell \in [k]$ and every $S \subseteq N$ with $|S| \geq q_\ell$,

$$\min_{i \in S} \Delta_\ell(i, X) \leq r_X(S).$$

Thus, any group large enough to deserve ℓ representatives and cohesive around some unselected candidate has at least one member with ℓ selected representatives within the corresponding radius. Moreover, mEJR+ strengthens mPJR+: for every unselected c, $r_X(S) \leq \max_{i \in S} \Delta(i, c)$, so the representatives guaranteed by mEJR+ also satisfy the mPJR+ requirement. Despite being stronger, mEJR+ can be verified much more efficiently.

Binary mEJR+ verification tells us whether a slate satisfies the axiom, but can provide a poor reward for policy optimization. First, unlike mPJR+, mEJR+ need not be satisfiable on every instance, in

[2]This is the strongly polynomial runtime bound. Using a weakly polynomial submodular-minimization algorithm instead gives $O!\left(n^5 k \log(nk) + n^5 \log^{O(1)}(nk)\right)$ time.

[3]Kraiczy et al. (2026, Sec. 5.2) use the FPT verifier for evaluation; its exponential dependence on k requires in turn subsampling their larger population.

which case the binary reward is identically zero. Second, even when satisfying slates exist, they may have very small probability under the current policy. If

$$\varepsilon_\theta(p) := \Pr_{X\sim\pi_\theta(\cdot|p)}[X \text{ satisfies mEJR+}],$$

then a batch of B independent policy samples contains no satisfying slate with probability $(1-\varepsilon_\theta(p))^B$. Thus, when $\varepsilon_\theta(p)$ is small, many policy updates may contain no satisfying slate.

We therefore seek an efficiently verifiable *graded* signal that distinguishes violating slates by how close they are to satisfying mEJR+. We obtain such a signal by multiplicatively relaxing Definition 2.

Definition 3 (mEJR+ distance-approximation factor)**.** For a finite $\beta \geq 0$, a slate X is a β-distance-approximation to mEJR+ if, for every $\ell \in [k]$ and every $S \subseteq N$ with $|S| \geq q_\ell$,

$$\min_{i\in S} \Delta_\ell(i, X) \leq \beta\, r_X(S).$$

The *mEJR+ distance-approximation factor*, or simply *mEJR+ factor*, $\beta^\star(X)$ is the smallest non-negative β satisfying these inequalities. If no finite β satisfies them, define $\beta^\star(X) = +\infty$.

In the ratio formulas below, we interpret $0/0$ as 0, and $a/0$ as $+\infty$ when $a > 0$.

Thus, $\beta^\star(X)$ quantifies the multiplicative relaxation needed for X to satisfy mEJR+: smaller values indicate stronger proportional representation, and X satisfies exact mEJR+ if and only if $\beta^\star(X) \leq 1$. Using the continuous measure $\beta^\star(X)$ is particularly useful as a verifiable reward for RL training because it distinguishes violating slates according to how close they are to satisfying mEJR+. It also remains informative on instances where exact mEJR+ cannot be satisfied.

The factor also gives a quantitative comparison with mPJR+: every mPJR+ slate X satisfies $\beta^\star(X) \leq 3$. To see this, fix ℓ and an eligible group S, and choose an unselected candidate c attaining $r = r_X(S)$. By mPJR+, at least ℓ selected representatives x are each within distance r of some member $j_x \in S$. For every $i \in S$, the triangle inequality gives

$$\Delta(i, x) \leq \Delta(i, c) + \Delta(c, j_x) + \Delta(j_x, x) \leq 3r.$$

Hence every member of S has ℓ selected representatives within distance $3r$, which proves the bound.

The following theorem shows that the mEJR+ factor can be computed efficiently.

Theorem 1 (Efficient computation of the mEJR+ factor)**.** *Given a slate X of size k, its optimal mEJR+ distance-approximation factor $\beta^\star(X)$ can be computed in $O(n^2(\log n + k))$ time.*

Corollary 1 (Efficient verification of mEJR+)**.** *Given a slate X of size k, whether X satisfies mEJR+ can be determined in $O(n^2(\log n + k))$ time.*

2.1 Computing the mEJR+ Factor

For each $\ell \in [k]$, define

$$\beta^\star_\ell = \max_{\substack{S\subseteq N \\ |S|\geq q_\ell}} \frac{\min_{i\in S} \Delta_\ell(i, X)}{r_X(S)}.$$

Then $\beta^\star(X) = \max_{\ell\in[k]} \beta^\star_\ell$. Algorithm 1 computes each $\beta^\star_\ell$ without enumerating the eligible groups.

Fix ℓ. Let T list voters in nondecreasing order of $\Delta_\ell(i, X)$. Initially all voters are active, and the algorithm removes them in this order. For each unselected candidate c, let L_c list voters in nondecreasing order of distance from c. Break ties in both kinds of lists by simulation index.

At the beginning of a round, let $u = T[i_T]$ be the next voter to be removed. For each c, maintain a pointer i_c to the q_ℓth active voter in L_c. While at least q_ℓ active voters remain, the smallest radius containing q_ℓ active voters around an unselected candidate is

$$r^\star = \min_{c\in N\setminus X} \Delta(c, L_c[i_c]).$$

The round contributes the ratio $\Delta_\ell(u, X)/r^\star$. Let β_ℓ be the largest ratio contributed during this sweep. The algorithm returns $\max_{\ell\in[k]} \beta_\ell$.

Algorithm 1 mEJR+ factor

Require: Indexed population N with distances $\Delta(i,j) = d(z_i, z_j)$; slate $X \subseteq N$ with $|X| = k$ and $1 \leq k < n$

For each $c \in N \setminus X$, form L_c by sorting voters by distance from c.
Store each voter's position in each list L_c.
For each $u \in N$, form X_u by sorting the elements of X by distance from u.
Break all distance-order ties by simulation index.
$\beta \leftarrow 0$
for $\ell = 1, \ldots, k$ **do**
    $q_\ell \leftarrow \lceil \ell n / k \rceil$
    $T \leftarrow N$ sorted by increasing $\Delta_\ell(u, X) = \Delta(u, X_u[\ell])$.
    Mark all voters as active.
    $i_T \leftarrow 1$; $\beta_\ell \leftarrow 0$
    $i_c \leftarrow q_\ell$ for every $c \in N \setminus X$
    while $i_T \leq n$ **do**
        $C_{\text{valid}} \leftarrow \{c \in N \setminus X : i_c \neq \text{None}\}$
        if $C_{\text{valid}} = \varnothing$ **then**
            break
        end if
        $u \leftarrow T[i_T]$
        $r^\star \leftarrow \min_{c \in C_{\text{valid}}} \Delta(c, L_c[i_c])$
        $a \leftarrow \Delta_\ell(u, X)$
        if $r^\star > 0$ **then**
            $\beta_\ell \leftarrow \max\{\beta_\ell, a / r^\star\}$
        else if $a > 0$ **then**
            return $+\infty$
        end if
        for each $c \in C_{\text{valid}}$ **do**
            if the stored position of u in L_c is at most i_c **then**
                Advance i_c to the next active voter after its current position in L_c, or set $i_c \leftarrow \text{None}$ if no such voter remains.
            end if
        end for
        Mark voter u as inactive.
        $i_T \leftarrow i_T + 1$
    end while
    $\beta \leftarrow \max\{\beta, \beta_\ell\}$
end for
return β

Pointer invariant. At the beginning of each inner-loop iteration, $L_c[i_c]$ is the q_ℓth active voter in L_c. If fewer than q_ℓ voters remain active, $i_c = \text{None}$.

Initially the invariant holds because all voters are active and $i_c = q_\ell$. When a voter is removed, a pointer remains unchanged if that voter occurs after it. Otherwise, advancing the pointer to the next active voter restores the invariant, or sets it to None if no such voter remains.

Lemma 1 (Runtime)**.** *The runtime of Algorithm 1 is* $O(n^2(\log n + k))$.

Proof Sketch. Preprocessing takes $O(n^2 \log n)$ time to sort, for every $c \in \mathcal{N}$, the voters by distance from c. The lists X_u can similarly be constructed in $O(nk \log k) \subseteq O(n^2 \log n)$ time.

Fix an outer-loop iteration ℓ. Sorting the voters by $\Delta_\ell(u, X)$ takes $O(n \log n)$. The inner loop has at most n iterations, since one voter is deactivated in each iteration. In every iteration, scanning all centers to determine C_{valid}, find r^*, and test which pointers must be updated costs $O(n)$. Moreover, for each center c, the pointer i_c only moves forward through L_c, so all pointer advances over the entire inner loop take $O(n^2)$ time in total. Thus, one outer-loop iteration costs $O(n^2)$.

Summing over the k values of ℓ gives $O(kn^2)$ time after preprocessing, and hence a total running time of $O(n^2(\log n + k))$. □

Proof of Theorem 1. The running-time bound follows from Lemma 1. For correctness, fix $\ell \in [k]$. We analyze the version of the sweep that records an infinite ratio and continues, rather than returning immediately; either version has the same final maximum.

First consider any round, with current voter u, and choose an unselected candidate c attaining $r^\star$. Let S consist of the q_ℓ closest active voters to c. Every member of S is active, so $\min_{i \in S} \Delta_\ell(i, X) \geq \Delta_\ell(u, X)$. Also, every member of S lies within distance $r^\star$ of c, so $r_X(S) \leq r^\star$. Therefore,

$$\frac{\Delta_\ell(u, X)}{r^\star} \leq \frac{\min_{i \in S} \Delta_\ell(i, X)}{r_X(S)} \leq \beta_\ell^\star.$$

Every contributed ratio is at most $\beta_\ell^\star$, giving $\beta_\ell \leq \beta_\ell^\star$.

Conversely, fix any group S with $|S| \geq q_\ell$. Consider the round in which its first member u is removed. Such a round occurs because the sweep cannot terminate while every member of S remains active. By the removal order, $\Delta_\ell(u, X) = \min_{i \in S} \Delta_\ell(i, X)$.

Choose an unselected candidate c attaining $r_X(S)$. Immediately before removing u, all members of S are active, so at least q_ℓ active voters lie within distance $r_X(S)$ of c. The pointer invariant therefore gives $r^\star \leq r_X(S)$. The ratio contributed in this round satisfies

$$\frac{\Delta_\ell(u, X)}{r^\star} \geq \frac{\min_{i \in S} \Delta_\ell(i, X)}{r_X(S)}.$$

Taking the maximum over eligible groups gives $\beta_\ell \geq \beta_\ell^\star$.

Both comparisons remain valid under the stated zero-denominator convention. Thus $\beta_\ell = \beta_\ell^\star$ for every ℓ, and the algorithm returns $\beta^\star(X)$. □

3 EXPERIMENTS

Our experiments evaluate whether training a router directly on $mEJR+$ to, given a prompt, select k simulations to route to improves representation of viewpoints at inference time. We compare RL-SCA with Predictive Route–then–Simulate(Kraiczy et al., 2026) and random selection in three domains, political viewpoints, personal advice, and book reviews, measuring representation on held-out prompts.

3.1 EXPERIMENTAL SET-UP

For each prompt p, we generate a response $A_i(p)$ from every simulation A_i and compute its viewpoint embedding $z_i(p) = \psi(A_i(p))$. These generated embeddings are used to compute training rewards and evaluate a selected slate, but are not inputs to RL-SCA. To evaluate the representativeness of the slate selected by each method, we use mPJR+ as used by (Kraiczy et al., 2026).

Methods. We compare the following routing approaches.

1. *RL-SCA (ours).* Our router maps a prompt embedding to a score for each simulation and selects the k highest-scoring simulations. It learns directly from the mEJR+ factor, without predicting individual response embeddings.
2. *Predictive Route–then–Simulate (PRS).* Following Kraiczy et al. (2026), this approach predicts a response embedding for each simulation from its context and the prompt, then runs SEAR on the predicted embeddings to select k simulations.
3. *Random.* This baseline selects k distinct simulations uniformly at random.

Domains and simulators. We evaluate on three domains of open-ended judgments: political viewpoints, personal advice, and book reviews. Our Remesh and Humanual Opinion datasets are constructed using the same data and simulation-generation pipelines as in Kraiczy et al. (2026). Additionally, we also consider Humanual Book, a book-review domain, using a different simulator.

1. *Political viewpoints (Remesh).* Remesh (Konya et al., 2023; 2025) is a collective response platform where participants express their viewpoints on contentious questions and vote on

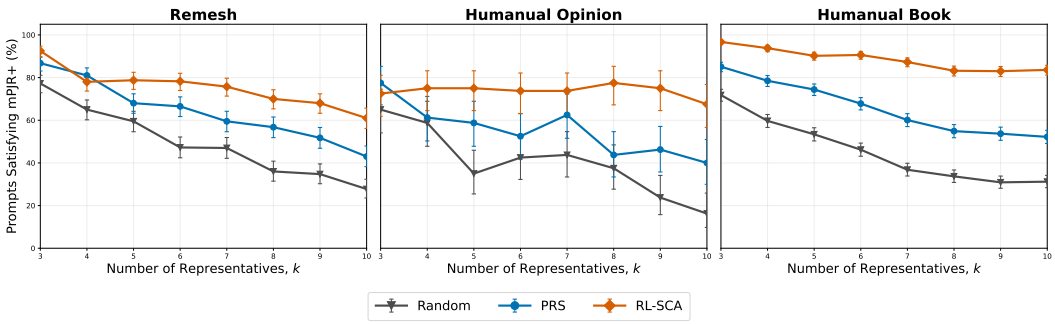

Figure 1: mPJR+ Satisfaction Rate by Simulation Selection Method.

whether they agree or disagree with other participants' responses. We use an open-source Remesh session in which 310 participants shared their views on the 2024 U.S. campus protests. Because the original session contains only four questions, we generate 1,000 additional prompts across ten related political and social topics for training and evaluation. We construct one simulation per participant using Llama-3.1-8B-Instruct (Grattafiori et al., 2024), conditioned on the participant's demographic profile, written responses, and agree/disagree votes from the original session. The model answers each new prompt in the role of that participant, without additional fine-tuning.

2. *Personal advice (Humanual Opinion).* On Reddit's "Am I the Asshole?" (AITA) subreddit, users describe personal situations and ask others to judge whether they were in the wrong. We use the `humanual-opinion` dataset collected by Wu et al. (2026), which contains 992 posts and responses from 4,567 users. These posts serve as our prompts, and the responses we simulate are individual users' judgments about the situations described. We use HumanLM as the simulator Wu et al. (2026), conditioning each response on the post and a summary of the user's Reddit history. Our experiments use a fixed subset of 300 users across all 992 posts.

3. *Book reviews (Humanual Book).* The book-review dataset from the Humanual benchmark introduced by Wu et al. (2026) is drawn from the Books category of Amazon Reviews 2023 and contains readers' written evaluations of books, together with a summary of their prior reviews. We use the supplied book prompts and summaries with `cmu-lti/osim-8b` to generate reviews for a fixed population of 103 users appearing in the source test split. We retain 3,431 prompts with a generated response from every selected user.

Further details on dataset construction, filtering, and prompt splits are given in Appendix D.

Viewpoint embeddings and router training. We use the preference embedding model of Blair et al. (2026) to embed prompts and responses. The router is a multilayer perceptron with two hidden layers. We train it using REINFORCE with reward $R_p(X) = -\log \beta_p^\star(X)$, where $\beta_p^\star(X)$ is the mEJR+ factor of the sampled slate in the generated response-embedding space. We examine the distribution of this training signal on held-out prompts in Section D.4. Training uses $k = 10$, and checkpoints are selected by validation mEJR+ satisfaction at $k = 10$. We evaluate deterministic top-k selection on held-out prompts for $k \in \{3, \ldots, 10\}$. Dataset construction, architecture, optimization, and model-selection details are given in Appendix D.

3.2 Results

Figure 1 shows the percentage of held-out prompts for which each routing method satisfies mPJR+ as the number of selected representatives k varies. Across all three domains, the main advantage of RL-SCA emerges at larger slate sizes, where the random baseline shows that the problem becomes more demanding because random slates are increasingly less likely to satisfy mPJR+. At $k = 10$, RL-SCA improves over Predictive Route–then–Simulate (PRS) substantially across the three datasets and the gains for most k are comparable in terms of magnitude to the relative improvement already obtained by using PRS over random selection. For example, on Humanual Book, mPJR+

satisfaction increases from approximately 53.7% under PRS to 83% under RL-SCA, an almost **30 percentage-point** improvement; the corresponding rate for random selection is 30.9%, roughly 23% percentage points behind PRS. We observe similarly large improvements on Remesh (43% $\rightarrow$ 61%) and Humanual Opinion (40% $\rightarrow$ 67.5%).

Notably, the router is trained only on slates of size $k = 10$, rather than training a separate router for each slate size. Nevertheless, the ranking it learns for each prompt also produces highly representative selections for smaller k. Because top-k selection simply takes progressively longer prefixes of this single learned ranking, this suggests that RL-SCA learns a useful ordering of simulated responses rather than a solution specialized to selecting exactly ten of them. This property is particularly useful for SAGE applications in which the number of viewpoints surfaced to a user may vary: a system can rank simulations once and select an appropriate prefix according to the desired response budget.

Satisfaction generally decreases as k increases. One possible explanation is that larger slates require the router to represent progressively finer-grained structure in the population: as k grows, smaller groups become entitled to representation under the proportionality criterion. Such prompt-specific distinctions may be harder to infer from the prompt alone than the broad viewpoint structure sufficient at smaller k. The decline could also reflect limitations of the router architecture or optimization procedure, and we leave disentangling these effects to future work.

Although, following prior work, our primary evaluation metric is exact mPJR+ satisfaction, RL-SCA is trained using the continuous mEJR+ factor $\beta_p^\star(X)$. We therefore provide a complementary analysis in Section D.4: in addition to exact mEJR+ satisfaction, we report the distribution of $\beta_p^\star(X)$ across held-out prompts. This distinguishes mild from severe violations and shows how much multiplicative relaxation is required when a selected slate does not satisfy exact mEJR+.

4 Discussion

Building on the formalization of scalable and representative routing for simulation-augmented generation (SAGE) by Kraiczy et al. (2026), we proposed RL-SCA as an alternative approach to representative routing in SAGE. Rather than first predicting a viewpoint embedding for every simulation for a given prompt and subsequently applying a proportional clustering algorithm, RL-SCA learns the routing decision end-to-end from a reward based on the continuous relaxation of a social-choice axiom. This is attractive because (1) it removes population-wide embedding prediction and clustering from inference, and (2) it substantially improves proportional representation on held-out prompts.

More broadly, our results show that social-choice guarantees can serve not only as post-hoc evaluation criteria, but also as learning objectives for representative routing across prompts. The mEJR+ factor enables RL-SCA because unlike the other proportionality objectives in the literature, it admits quantitative evaluation efficient enough for reinforcement learning; extending RL-SCA to other axioms will require similarly tractable quantitative verifiers.

We view the present results as a proof of concept rather than sufficient evidence for deployment; practical use would require substantially more extensive robustness evaluation. Our current instantiation of RL-SCA is deliberately simple and instantiates the router as a small MLP trained for a fixed population and a single slate size, with one output per simulation. Future work could train jointly over multiple slate sizes, use richer prompt conditioning and adopt architectures inducing more expressive set-selection policies, such as autoregressive selection, to capture prompt-dependent complementarity and redundancy between selected simulations. An important direction in the context of SAGE (Milli et al., 2026) is to develop routers that can select representatives from arbitrary subpopulations and more generally adapt to simulation populations that grow or change over time.

References

Haris Aziz, Markus Brill, Vincent Conitzer, Edith Elkind, Rupert Freeman, and Toby Walsh. Justified representation in approval-based committee voting. *Social Choice and Welfare*, 48(2):461–485, 2017.

Haris Aziz, Edith Elkind, Shenwei Huang, Martin Lackner, Luis Sánchez-Fernández, and Piotr Skowron. On the complexity of extended and proportional justified representation. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pp. 902–909, 2018.

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*, 2024.

Tuva Bardal, Markus Brill, David McCune, and Jannik Peters. Proportional representation in practice: Quantifying proportionality in ordinal elections. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 13581–13588, 2025. doi: 10.1609/aaai.v39i13.33483.

Carter Blair, Ariel D. Procaccia, and Milind Tambe. Embeddings for Preferences, Not Semantics. 2026.

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 '23, pp. 301, New York, NY, USA, 2023a. Association for Computing Machinery. ISBN 9798400701047. doi: 10.1145/3580507.3597785. URL https://doi.org/10.1145/3580507.3597785.

Markus Brill and Jannik Peters. Robust and verifiable proportionality axioms for multiwinner voting. In *Proceedings of the 24th ACM Conference on Economics and Computation*, pp. 301. Association for Computing Machinery, 2023b. ISBN 9798400701047. doi: 10.1145/3580507.3597785.

Chris Cameron, Jason Hartford, Taylor Lundy, and Kevin Leyton-Brown. The perils of learning before optimizing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 3708–3715, 2022.

Ioannis Caragiannis, Evi Micha, and Nisarg Shah. Proportional fairness in non-centroid clustering. In *Advances in Neural Information Processing Systems*, volume 37, pp. 19139–19166, 2024. doi: 10.52202/079017-0605.

Alfonso Cevallos and Alistair Stewart. A verifiably secure and proportional committee election rule. In *Proceedings of the 3rd ACM Conference on Advances in Financial Technologies*, pp. 29–42, 2021. doi: 10.1145/3479722.3480988.

Xingyu Chen, Brandon Fain, Liang Lyu, and Kamesh Munagala. Proportionally fair clustering. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1032–1041. PMLR, 2019.

DeepSeek-AI. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645:633–638, 2025. doi: 10.1038/s41586-025-09422-z.

Priya Donti, Brandon Amos, and J. Zico Kolter. Task-based end-to-end model learning in stochastic optimization. In *Advances in Neural Information Processing Systems*, volume 30, pp. 5484–5494, 2017.

Adam N. Elmachtoub and Paul Grigas. Smart "predict, then optimize". *Management Science*, 68 (1):9–26, 2022. doi: 10.1287/mnsc.2020.3922.

Shangbin Feng, Taylor Sorensen, Yuhan Liu, Jillian Fisher, Chan Young Park, Yejin Choi, and Yulia Tsvetkov. Modular Pluralism: Pluralistic Alignment via Multi-LLM Collaboration. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 4151–4171, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.240. URL https://aclanthology.org/2024.emnlp-main.240/.

Sara Fish, Paul Gölz, David C. Parkes, Ariel D. Procaccia, Gili Rusak, Itai Shapira, and Manuel Wüthrich. Generative social choice. *Journal of the ACM*, 73(2):1–52, 2026.

Jillian Fisher, Ruth Elisabeth Appel, Chan Young Park, Yujin Potter, Liwei Jiang, Taylor Sorensen, Shangbin Feng, Yulia Tsvetkov, Margaret Roberts, Jennifer Pan, Dawn Song, and Yejin Choi. Position: Political Neutrality in AI Is Impossible — But Here Is How to Approximate It. In *Forty-second International Conference on Machine Learning Position Paper Track*, 2025. URL https://openreview.net/forum?id=H72JEXAPwo.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Yu He, Jeremy Vollen, and Edith Elkind. Check, Please: Verifiably Fair Clustering, 2026. URL https://arxiv.org/abs/2605.12317.

Leon Kellerhals and Jannik Peters. Proportional Fairness in Clustering: A Social Choice Perspective. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=KsLX5pFpOs.

Andrew Konya, Lisa Schirch, Colin Irwin, and Aviv Ovadya. Democratic policy development using collective dialogues and AI. *arXiv preprint arXiv:2311.02242*, 2023.

Andrew Konya, Luke Thorburn, Wasim Almasri, Oded Adomi Leshem, Ariel Procaccia, Lisa Schirch, and Michiel Bakker. Using collective dialogues and AI to find common ground between Israeli and Palestinian peacebuilders. In *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '25, pp. 312–333, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400714825. doi: 10.1145/3715275.3732022. URL https://doi.org/10.1145/3715275.3732022.

Sonja Kraiczy, Smitha Milli, Ratip Emin Berker, Avinandan Bose, Brandon Amos, Jamelle Watson-Daniels, Maximilian Nickel, Edith Elkind, and Ariel D. Procaccia. Social choice foundations for simulation-augmented generation. In *Advances in Neural Information Processing Systems*, volume 39, 2026. To appear.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tülu 3: Pushing frontiers in open language model post-training. In *Second Conference on Language Modeling*, 2025.

Smitha Milli, Ratip Emin Berker, Sonja Kraiczy, Claudia Shi, Jack Kussman, Avinandan Bose, Edith Elkind, Himaghna Bhattacharjee, Ariel Procaccia, and Maximilian Nickel. For Questions of Ought, AI Could Use Some SAGE Advice. 2026.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Mrinank Sharma, Miles McCain, Raymond Douglas, and David Duvenaud. Who's in Charge? Disempowerment Patterns in Real-World LLM Usage. *arXiv preprint arXiv:2601.19062*, 2026.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

Taylor Sorensen, Jared Moore, Jillian Fisher, Mitchell 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'24. JMLR.org, 2024.

Shirley Wu, Evelyn Choi, Arpandeep Khatua, Zhanghan Wang, Joy He-Yueya, Tharindu Cyril Weerasooriya, Wei Wei, Diyi Yang, Jure Leskovec, and James Zou. Humanlm: Simulating users with state alignment beats response imitation. 2026. URL https://humanlm.stanford.edu/.

A LLM USAGE

We used generative AI tools, including OpenAI ChatGPT, to assist with manuscript editing and identifying potentially relevant literature. OpenAI Codex and Claude Code was used to assist with implementing the experimental pipeline. The experimental pipeline also uses AI models as experimental components, as detailed in the relevant sections of the main text and appendix. All AI-assisted code was reviewed and tested by the authors, and all AI-assisted manuscript revisions and references were independently checked. The authors take full responsibility for the final content.

B RELATED WORK

Social choice. An influential line of research in social choice formalizes proportional representation in committee elections (Aziz et al., 2017; 2018). Broadly, these notions require sufficiently large and cohesive groups of voters to receive representation commensurate with their size. Analogous guarantees have been studied in clustering, where voters and representatives lie in a metric space (Chen et al., 2019; Aziz et al., 2024; Kellerhals & Peters, 2024). We adopt this geometric perspective as well. A related line of work studies whether proportionality guarantees can be verified efficiently. Motivated by validator selection in proof-of-stake blockchains, Cevallos & Stewart (2021) develop efficiently verifiable proportionality guarantees, and subsequent work treats efficient verifiability itself as a design criterion for proportionality axioms (Brill & Peters, 2023a).

Beyond binary verification, prior work has proposed quantitative notions of proportionality in ordinal elections (Bardal et al., 2025) and approximate auditing procedures for proportional fairness in non-centroid clustering (Caragiannis et al., 2024). These works differ from ours in both setting and objective: Bardal et al. quantify proportionality of election outcomes, while Caragiannis et al. estimate the approximation quality of a given clustering up to a constant factor. In contrast, we give an exact polynomial-time algorithm for the tight mEJR+ distance-approximation factor in the centroid-based metric setting, and use this quantity directly as a reward for reinforcement learning.

Pluralistic alignment. Since we build on the framework by Kraiczy et al. (2026), our work is closely related to work advocating for exposing users to a plurality of viewpoints, including Overton pluralism (Sorensen et al., 2024), reasonable pluralism (Fisher et al., 2025), and work on mitigating disempowerment (Sharma et al., 2026). Our work is also related to generative social choice (Fish et al., 2026), which aims to generate a slate of viewpoints that is formally representative of a population. These approaches use discriminative and generative queries as building blocks; in the SAGE setting, however, realizing the latter requires repeated LLM calls, which conflicts with the goal of querying only a small number of simulations at inference time. Modular pluralism (Feng et al., 2024) similarly queries multiple "community LLMs" to surface diverse perspectives, but uses a fixed, pre-specified collection of models rather than dynamically routing over a larger population.

Reinforcement learning with verifiable rewards. Reinforcement learning with verifiable rewards (RLVR) uses rewards that can be computed automatically from a model's output, avoiding the need to learn a separate reward model. Lambert et al. (2025) introduced the term RLVR and applied it to domains with automatically checkable outcomes, including mathematical reasoning and verifiable instruction following. Related work has shown that reinforcement learning with automatically checkable rewards can substantially improve reasoning on mathematics and code (DeepSeek-AI, 2025); Shao et al. (2024) introduced GRPO, the policy-optimization method subsequently used in this line of work. RL-SCA extends this general idea to representative routing, where there is no single correct output: instead, the reward is obtained by efficiently verifying how well a selected slate satisfies a social-choice guarantee.

C DETAILS OMITTED FROM SECTION SECTION 2

Lemma 1 (Runtime)**.** *The runtime of Algorithm 1 is* $O(n^2(\log n + k))$.

Proof. We assume that each pairwise distance is available in constant time.

We first analyze the preprocessing. For each $c \in \mathcal{N}$, we construct the list L_c of voters in $\mathcal{N}$ sorted by increasing distance from c. This takes $O(n^2 \log n)$ time over all c. Similarly, for each $u \in \mathcal{N}$, we

construct the list X_u of elements of X sorted by increasing distance from u, which takes $O(nk \log k)$ time. Since $k \leq n$, the total preprocessing time is

$$O(n^2 \log n + nk \log k) = O(n^2 \log n).$$

Now fix an iteration of the outer loop, corresponding to some $\ell \in \{1, \ldots, k\}$, and let $q = \lceil \frac{\ell n}{k} \rceil$. The algorithm first sorts the voters $u \in \mathcal{N}$ by

$$\Delta_\ell(u, X) = \Delta(u, X_u[\ell]).$$

Since the lists X_u have already been sorted, each value $\Delta_\ell(u, X)$ can be read in $O(1)$ time. Thus constructing and sorting T takes $O(n \log n)$ time. Marking all voters as active, initializing i_T, β_ℓ, and setting $i_c \leftarrow q$ for every $c \in \mathcal{N}$ takes $O(n)$ additional time.

We next bound the total time spent in the while loop for this fixed ℓ. The pointer i_T increases by one in every non-terminal iteration, and therefore the while loop has at most n non-terminal iterations.

In each iteration, the set

$$C_{\text{valid}} = \{c \in N \setminus X : i_c \neq \text{None}\}.$$

can be found by scanning all $c \in \mathcal{N}$, taking $O(n)$ time. The value

$$r^* = \min\{\Delta(c, L_c[i_c]) : c \in C_{\text{valid}}\}$$

can also be computed by scanning all valid centers, again taking $O(n)$ time. Therefore these operations contribute $O(n^2)$ total time over the while loop.

It remains to bound the cost of updating the pointers i_c. For each $c \in C_{\text{valid}}$, we need to test whether the newly deactivated voter $T[i_T]$ appears at or before position i_c in L_c. This can be tested in $O(1)$ time, either by using precomputed ranks in the lists L_c, or equivalently by comparing the lexicographic distance-order keys. Hence these membership tests cost $O(n)$ per while-loop iteration and $O(n^2)$ total for this fixed ℓ.

Whenever the test succeeds for some c, the pointer i_c is advanced to the next active voter after its current position in L_c, or set to None if no such voter exists. For each fixed c, the pointer i_c only moves forward through the list L_c. Hence, over all iterations of the while loop, i_c advances at most n times. Summing over all $c \in \mathcal{N}$, the total time spent advancing pointers is $O(n^2)$ for this fixed ℓ.

Thus, for a fixed ℓ, the while loop costs $O(n^2)$ time, and the sorting and initialization before the while loop cost $O(n \log n)$. Therefore one outer-loop iteration costs

$$O(n^2 + n \log n) = O(n^2).$$

There are k iterations of the outer loop, so the total time after preprocessing is $O(kn^2)$.

Combining this with the preprocessing cost, the total running time is

$$O(n^2 \log n) + O(kn^2) = O(n^2(\log n + k)).$$

□

D Experimental Details

D.1 Datasets and simulation

For each dataset, we use a fixed simulation population across all prompts. Every retained prompt has a response from every member of that population. Table 1 summarizes the resulting datasets.

D.1.1 Remesh

Prompt construction. The Remesh campus-protest study contains 310 participants, their attributes, written responses, and agree/disagree votes. We use ten predefined political and social topics and generate 100 questions per topic with Claude Opus 4.6, giving 1,000 prompts. The recorded model identifier is `claude-4-6-opus`, and question generation uses temperature 0.

Dataset	Users	Train	Validation	Test	Responses
Remesh	310	500	100	400	310,000
Humanual Opinion	300	812	100	80	297,600
Humanual Book	103	2,000	431	1,000	353,393

Table 1: Prompt splits and fixed simulation populations. The train, validation, and test columns count prompts; the final column counts generated responses across all three splits.

Simulation. For every participant–prompt pair, Llama-3.1-8B-Instruct generates a two- or three-sentence response while role-playing the participant. Its context contains the participant's attributes and their original Remesh responses and votes. Response generation uses temperature 0 and seed 42, producing $310{,}000$ simulated responses.

Split. We partition the $1{,}000$ prompts into 500 training, 100 validation, and 400 test prompts using seed 42. All 310 participants remain in each split.

D.1.2 HUMANUAL OPINION

Prompts and simulation. Humanual Opinion contains 992 real Reddit AITA posts and persona descriptions for $4{,}567$ users. We use these posts directly as prompts; they are not generated for this experiment. The checkpoint `snap-stanford/humanlm-opinion` receives a user persona and a post and generates a comment in that user's voice. Generation uses temperature 0.4, a maximum of $1{,}024$ new tokens, and a four-token no-repeat constraint.

Population selection. After filtering blank generations, we retain $4{,}508$ users with responses to every post. We sort these users, shuffle them using seed 42, and select 300. The same users serve as both candidates and voters for every prompt; we do not subsample the population separately for each prompt. The resulting dataset contains $992 \times 300 = 297{,}600$ responses.

Split. We preserve the original 80-post test set. We pool the original training and validation posts and repartition them into 812 training and 100 validation posts using seed 42. Test prompts are used only for final evaluation.

D.1.3 HUMANUAL BOOK

Prompts and population. Humanual Book contains product-review records with book prompts, user persona descriptions, and source training, validation, and test labels. We use the 103 users appearing in the source test split as the fixed population for all three splits. Each user's supplied persona is shared across prompts. We extract the nonempty content of the first prompt message and deduplicate prompts by `post_id`.

Simulation. The post-trained `cmu-lti/osim-8b` model generates a response for each selected user–prompt pair. We use bfloat16 inference, temperature 0.7, top-p 1.0, no top-k truncation, and repetition penalty 1.0. The prompt and maximum response budgets are each $8{,}192$ tokens, with a model context of $17{,}408$ tokens. Overlong inputs are left-truncated using the model's chat template. We extract the response text from the `<response>` element.

Complete response grid and split. The initial 103-user dataset contains $16{,}516$ prompts and $1{,}701{,}108$ nonblank responses. Before constructing the experimental subset, we exclude prompts missing a response from any of the 103 users. We then select $2{,}000$ training and $1{,}000$ test prompts using seed 42 and retain 431 validation prompts. The resulting complete grid contains $3{,}431$ prompts and $3{,}431 \times 103 = 353{,}393$ responses.

D.2 VIEWPOINT EMBEDDING MODEL

We encode both prompts and generated responses using `cartgr/embeddings-for-preferences-st5-xl`, the preference embedding model of Blair et al. (2026). All embeddings are normalized 768-dimensional vectors. For a prompt p,

Dataset	Population n	Hidden width h	Parameters
Remesh	310	64	73,526
Humanual Opinion	300	128	153,644
Humanual Book	103	64	60,071

Table 2: Router architectures. Parameter counts exclude the prompt embedding model and simulators.

Dataset	h	η	Samples	Epoch	Val. (%)
Remesh	64	3×10^{-4}	8	29	66
Humanual Opinion	128	10^{-4}	8	82	76

Table 3: Selected training configurations. Samples is the number of sampled slates per prompt. Epoch is the selected checkpoint epoch, and Val. is deterministic validation mEJR+ satisfaction at $k = 10$.

the generated response embeddings $(z_i(p))_{i=1}^n$ define the population against which sampled and selected slates are evaluated. The router receives the prompt embedding, not the generated response embeddings.

D.3 RL-SCA ARCHITECTURE AND TRAINING

Architecture. The router is a ReLU multilayer perceptron with architecture

$$768 \longrightarrow h \longrightarrow h \longrightarrow n,$$

where n is the number of simulations in the fixed population. Its output contains one score per simulation. Table 2 gives the hidden widths and parameter counts for the three datasets.

Sampling and reward. For each training prompt, the router outputs one logit per available simulation. We draw eight slates of ten distinct simulations via sequential sampling without replacement: at each step, a simulation is sampled from the softmax distribution over the remaining logits and then removed. Each slate is evaluated against the full fixed population using the mEJR+ factor, with reward

$$R_p(X) = -\log \beta_p^\star(X).$$

For the bth sample, the advantage subtracts the mean reward of the other seven samples for the same prompt:

$$A_{p,b} = R_p(X_{p,b}) - \frac{1}{7} \sum_{b' \neq b} R_p(X_{p,b'}).$$

The log probability of a sampled slate is computed from the sequence of conditional sampling probabilities used to draw its members without replacement. We use these advantages for REINFORCE updates. At evaluation time, we do not sample: the router returns the k distinct highest-scoring simulations.

Optimization. We use AdamW with a batch size of 16 prompts and clip gradients at norm 1.0. All reported hyperparameter sweeps use training seed 0. The checkpoint-selection criterion is deterministic validation mEJR+ satisfaction at $k = 10$.

Hyperparameter selection. For Remesh, we sweep hidden width $h \in \{64, 128\}$, learning rate $\eta \in \{3 \times 10^{-5}, 10^{-4}, 3 \times 10^{-4}\}$, and the number of sampled slates per prompt $B \in \{8, 16, 32\}$. Runs are scheduled for 50 epochs. Twelve of the 18 runs complete; the six runs with $B = 32$ stop after 20–24 epochs.

For Humanual Opinion, we sweep $h \in \{64, 128\}$ and $\eta \in \{10^{-4}, 3 \times 10^{-4}\}$, with $B = 8$ samples per prompt. All four runs complete 100 epochs. Table 3 reports the selected configurations for Remesh and Humanual Opinion.

D.4 Evaluation of mEJR+

In the main body, we evaluate the selected slates on held-out prompts using the generated response embeddings of the full fixed population and report the percentage of prompts on which the slate satisfies mPJR+ for $k \in \{3, \ldots, 10\}$. Here we also report the satisfaction rate of mEJR+, the axiom used to construct the training reward, determined by whether $\beta_p^\star(X) \leq 1$. RL-SCA uses deterministic, distinct top-k selections, while random selection draws k distinct simulations uniformly. The figures include 95% binomial confidence intervals where reported. Figure 2 reports both mPJR+ and mEJR+ satisfaction rates for the same selected slates. Since mEJR+ implies mPJR+, the mEJR+ satisfaction rate cannot exceed the corresponding mPJR+ satisfaction rate. The improvement of RL-SCA over PRS is generally at least as large for mEJR+ as for mPJR+.

This comparison should be interpreted in light of the objectives used by the two methods. PRS applies SEAR to the predicted viewpoint geometry. SEAR is guaranteed to satisfy mPJR+, but need not return an mEJR+ slate even when one exists. RL-SCA, in contrast, is trained directly using the mEJR+ factor. We therefore report mPJR+ as the primary metric in the main text, both for comparability with prior work and because it is not the objective directly optimized by RL-SCA.

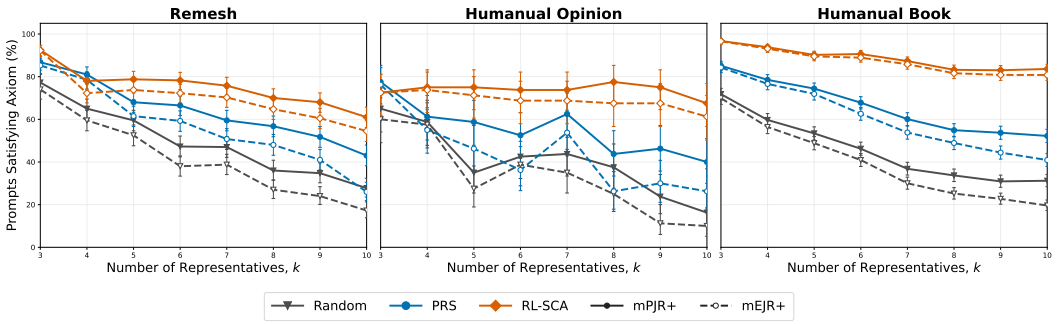

Figure 2: mPJR+ and mEJR+ satisfaction rates by simulation selection method.

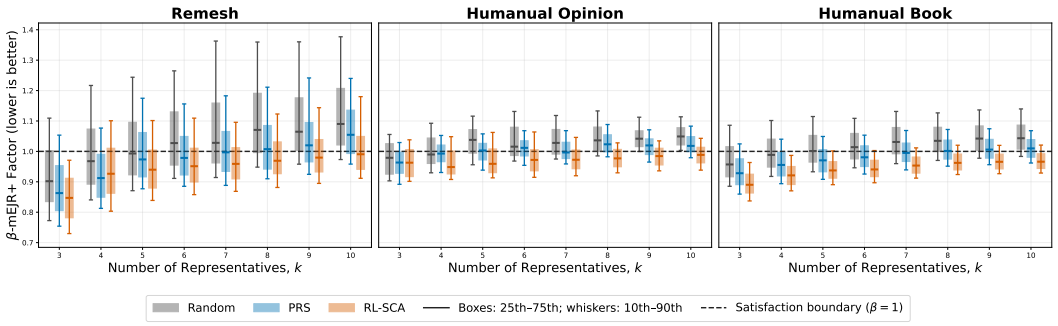

Figure 3: Distribution of the mEJR+ factor by simulation selection method. Lower values indicate stronger proportional representation, and $\beta^\star \leq 1$ corresponds to exact mEJR+ satisfaction.

Results. Figure 3 shows the distribution of the mEJR+ factor across held-out prompts. Unlike binary mEJR+ satisfaction, the factor also distinguishes between violating slates according to the severity of the violation. For example, $\beta^\star = 1.35$ means that satisfying the mEJR+ inequalities for that slate requires a 35% multiplicative relaxation of the distance bound.

At $k = 10$, RL-SCA substantially improves test-set mEJR$^+$ satisfaction over both baselines. On Remesh, Humanual Opinion, and Humanual Book, RL-SCA achieves satisfaction rates of 54.5%, 61.25%, and 80.8%, respectively, compared with 17.25%, 10%, and 19.60% for Random and 26%, 26.25%, and 40.9% for PRS. This corresponds to improvements over PRS of 28.5, 35, and 39.90 percentage points, respectively.

The mEJR+ factor is already a worst-case measure within each prompt, since it takes the maximum over all values of ℓ and all eligible groups. It is therefore also informative to consider the largest

factor observed across prompts. At $k = 10$, the largest factor on Remesh decreases from 2.914 for Random to 2.653 for PRS and 1.887 for RL-SCA. For Humanual Opinion, the corresponding values are 1.296, 1.204, and 1.091. For Humanual Book, they are 1.51, 1.244, and 1.212, respectively. Thus, although all three methods have at least one prompt that violates exact mEJR+ at $k = 10$, RL-SCA reduces the largest observed violation across all three datasets.