Generating Fair Consensus Statements with Social Choice on Token-Level MDPs

Carter Blair
Harvard University
Cambridge, USA
carterblair@g.harvard.edu

Kate Larson
University of Waterloo
Waterloo, Canada
kate.larson@uwaterloo.ca

ABSTRACT

Current frameworks for aggregating free-form text-based opinions with large language models lack the inherent structure needed to provide meaningful fairness guarantees. To address this, we model the task as a multi-objective, token-level Markov Decision Process (MDP), where each objective corresponds to an agent's preference. Each agent's token-level reward is induced by its policy (e.g., a personalized language model). Such policies implicitly define optimal Q-functions, thus enabling stepwise reward computation without an explicit value function [18]. This MDP formulation yields a formal structure that can be analyzed with tools from social choice theory. We first give a stochastic generation policy that is guaranteed to lie in the ex-ante core. It is derived from a distribution over complete statements that maximizes Nash welfare, extending core stability from cooperative game theory and voting to text generation. Second, for a single consensus statement, we target egalitarian welfare and use search within the MDP. Empirically, this search produces statements with improved worst-case agent alignment compared with baselines, including the Habermas Machine [24]. Our code is available here and supplementary material (the appendix) is available here.

KEYWORDS

Generative Social Choice, Guided Decoding, Fairness

ACM Reference Format:
Carter Blair and Kate Larson. 2026. Generating Fair Consensus Statements with Social Choice on Token-Level MDPs. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 9 pages. https://doi.org/10.65109/NZPR5925

1 INTRODUCTION

Social choice theory has traditionally studied how to aggregate preferences over predefined alternatives. Large language models (LLMs) make it possible to aggregate free-form verbal opinions into collective textual outputs, which removes the constraint of a fixed agenda. Importantly, this flexibility can reduce the agenda-setting power of organizers, since participants need not choose from a preset list. However, ensuring provable fairness is difficult: complex training and design choices yield an opaque, irregular LLM structure, which in turn complicates the formalization of specific fairness criteria during generation. For this reason, previous approaches have treated the generation process as a black box, applying various fairness measures post-hoc. For instance, in the Habermas Machine [24], statements are first generated, and fairness is then pursued through a voting procedure applied to these statements, which were not themselves generated with an explicitly fair mechanism. Similarly, the Generative Social Choice method [10] prompts an LLM to maximize a given objective and assumes that the response truly maximizes the objective. These strategies, though aimed at fairness in free-form opinion aggregation, can cede a new form of agenda control to the LLM. When the model is asked to produce text that satisfies a broad fairness objective, its interpretation and implementation of that objective, the trade-offs it chooses, and the aspects of opinions it elevates remain opaque. This effectively allows the LLM to shape the solution space. As such, these methods risk overlooking biases embedded within the generation process itself, which are known to exist [9].

We address this gap by modeling consensus statement generation as a token-level Markov Decision Process (MDP). Each agent i's viewpoint is represented by a policy π_i, which assigns likelihoods $\pi_i(s, a)$ to token choices given the current prefix s. Following Rafailov et al. [18], who show that policies implicitly define optimal Q-functions, our agent policies, π_i, determine rewards $r_i(s, a)$ (e.g., $r_i^{\log}(s, a) = \beta \log \pi_i(s, a)$) at each generation step. A primary advantage of this reward formulation is that it avoids personalized value functions, which are known to be challenging to train and apply effectively [13]. This MDP structure provides a formal basis for integrating fairness principles directly into the construction of the consensus statement.

Within this MDP framework, we develop two approaches that leverage existing notions of fairness from social choice theory, namely the *ex-ante core* and *egalitarian welfare (EW)*, which we argue are compelling notions of fairness in the context of consensus statement generation. First, to achieve an outcome in the ex-ante core, we propose a stochastic generation policy π^*. This policy is derived by optimizing a distribution over complete statements to maximize proportional fairness (Nash Welfare), a process known to yield core membership. For consensus generation, the core is a highly desirable stability concept: a lottery in the core ensures that no coalition of agents could unilaterally deviate and achieve an alternative lottery that all its members prefer, given their proportional influence, implying agreement, or even consensus, over the randomized outcome. Second, when a single consensus statement is desired, we target the maximization of EW, seeking the best outcome for the least satisfied agent, which aligns with the idea that a consensus statement should be agreeable to all parties. We introduce constructive search algorithms (finite lookahead and beam search) that optimize this EW objective directly within the

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/NZPR5925

MDP. This offers a transparent and analyzable generation mechanism distinct from methods reliant on high-level prompting or post-hoc voting.

Our main contributions are:

(1) A formal token-level MDP framework for fair consensus generation where agent rewards are derived from their language model policies.
(2) A method to derive a stochastic generation policy that is provably in the ex-ante core, ensuring proportional fairness and stability.
(3) The development and empirical validation of search algorithms that, by optimizing egalitarian welfare within the MDP, generate single consensus statements with improved worst-case agent alignment compared to methods that do not leverage this token-level structure or search.

Through these contributions, we hope to establish a new direction for methods seeking to generate consensus statements from open-ended verbal opinions with provable fairness guarantees.

2 RELATED WORK

Generative Social Choice. This field applies social choice principles to open-ended generation, such as creating text from diverse opinions [3, 10, 21, 24]. Unlike methods that aggregate preferences over predefined alternatives, Generative Social Choice (GSC) generates the alternatives themselves. For example, Tessler et al. [24] employ iterative critiques and voting on complete statements for consensus, while Fish et al. [10] prompt LLMs to directly maximize egalitarian welfare within a larger framework aimed at a form of proportional representation. Our work differs by embedding fairness into the token-by-token construction of a consensus statement via a multi-objective MDP, treating each token selection as a public decision [6]. This provides a more granular and verifiable mechanism than post-hoc evaluations or high-level prompting.

Randomized Social Choice. We also draw from randomized social choice, which studies lotteries over outcomes. Our stochastic generation policy, which maximizes Nash Welfare for proportional fairness, connects to this area. Maximizing Nash Welfare is known to yield outcomes in the 1-core [1, 7, 8]. We extend these findings to the sequential decision-making context of this paper.

Guided Decoding. Guided decoding techniques steer LLM generation towards desired attributes at inference time, often using search algorithms. Methods like PPO-MCTS [17] and VAS [14] use a value network to guide generation, while MOD [23] or COLLAB [5] combine or switch policies. Our approach also uses search but derives token-level rewards from agent policies. Further, we explicitly frame generation as planning in a multi-objective MDP to optimize social choice objectives (Proportional Fairness, Egalitarian Welfare), rather than relying on a single pre-trained value model or heuristic model combinations.

3 PROBLEM SETUP & PRELIMINARIES

We consider a setting with a finite set of agents $N = \{1, 2, \ldots, n\}$, each with a distinct opinion on a specific **Issue**. The goal is to generate a consensus text statement reflecting these perspectives fairly. The inputs to the process include descriptions of the issue, a policy representing each agent (which can be derived from free-form text expressing their opinion), and a **reference policy** (e.g., a base language model) that is used to propose tokens in the consensus statement.

Agent Policies. Each agent $i \in N$ is represented by a *policy* π_i, which assigns a likelihood $\pi_i(s, a) \in [0, 1]$ to each action a given the state s. Intuitively, $\pi_i(s, a)$ reflects how closely an action aligns with agent i's preference at state s. This policy could be an LLM fine-tuned (ideally with DPO [19]) or base policy prompted with agent i's viewpoint.

Token-Level MDP. We model text generation as a deterministic, discrete-time Markov Decision Process defined by the tuple $(S, A, T, \mathbf{R})$. Here, S is the state space of partial text sequences (prefixes), including initial s_0 and terminal states. A is the action space consisting of the token vocabulary plus a special end-of-sequence token $\langle\text{eos}\rangle$. T is the deterministic transition function where $T(s, a) = s \| a$ appends the chosen token; selecting $a = \langle\text{eos}\rangle$ leads to a terminal state representing a completed statement X. Finally, $\mathbf{R}$ represents the agent-specific reward functions. We define two types of rewards based on agent policies, serving different analytical purposes:

(1) **Log-Likelihood Reward:** $r_i^{\log}(s, a) = \beta \log \pi_i(s, a)$. This formulation aligns with implicit rewards in preference learning [18], where $\beta > 0$ is a scaling factor. This is non-positive and is suitable for additive utility accumulation along a path.
(2) **Likelihood Reward:** $r_i^{\text{prob}}(s, a) = \pi_i(s, a)$. This reward uses the direct probability, ensuring non-negativity ($r_i^{\text{prob}} \geq 0$), which is needed for social welfare functions involving products or ratios, such as Nash Welfare.

We denote by C the set of all possible complete paths (sequences ending in $\langle\text{eos}\rangle$) from s_0.

In practice, we assume that at each non-terminal state s, we only consider a finite set of B possible next tokens $A_B(s) \subseteq A$. This set could be the model's vocabulary or a subset chosen by a base language model giving us the B most likely next tokens. With this finite branching factor B and a maximum sequence length $L_{\max}$, the set C of complete paths is finite (bounded by $B^{L_{\max}}$).

This sequential token selection process naturally defines a tree structure rooted at s_0. Each edge represents choosing a token from $A_B(s_t)$, and each node represents a partial sequence s_t. To make this concrete, see panel B of Figure 1.

Agent Utilities. Given a completed sequence $X = (a_1, \ldots, a_\ell = \langle\text{eos}\rangle)$ corresponding to states $(s_0, s_1, \ldots, s_\ell)$, we define two corresponding utility functions for each agent i, derived from the respective reward types:

(1) **Additive Log-Utility:** Primarily used for evaluating single paths based on cumulative log-likelihood.

$$U_i^{\log}(X) = \sum_{t=1}^{\ell} r_i^{\log}(s_{t-1}, a_t) = \sum_{t=1}^{\ell} \beta \log \pi_i(s_{t-1}, a_t)$$
$$= \beta \log \left(\prod_{t=1}^{\ell} \pi_i(s_{t-1}, a_t) \right)$$

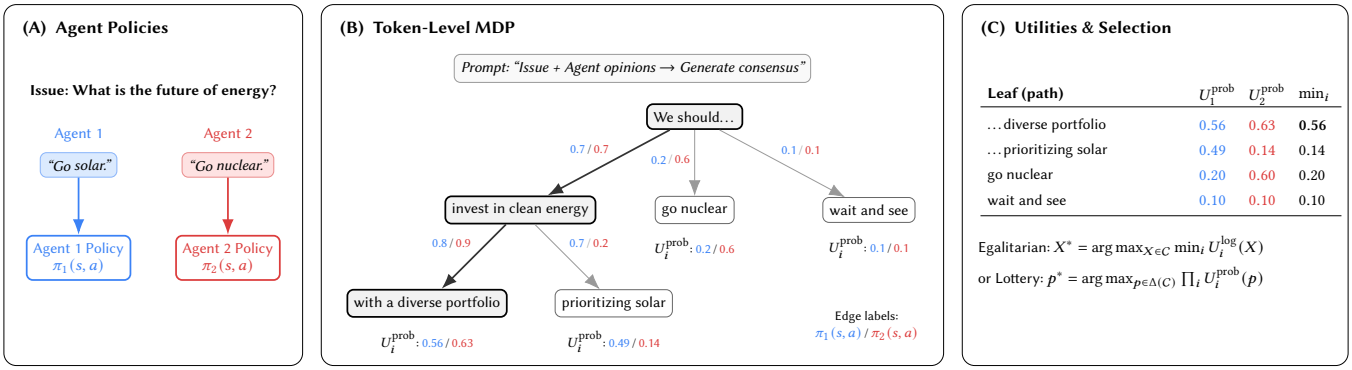

Figure 1: Toy example of the consensus generation pipeline. (A) Each agent holds a policy $\pi_i(s,a)$ over tokens, determined by its stated opinion. (B) A token-level MDP expands candidate consensus statements $X \in C$ as a tree; edge labels show each agent's policy probability (π_1/π_2) for that token/chunk, and leaf labels show the multiplicative probability utility $U_i^{\text{prob}}(X) = \prod_t \pi_i(s_{t-1}, a_t)$. The full prompt is given in Appendix F. (C) Each complete path defines a candidate with per-agent utility $U_i^{\text{prob}}(X)$. The egalitarian rule selects the candidate maximizing $\min_i U_i^{\log}(X)$; alternatively, a lottery $p \in \Delta(C)$ maximizes Nash welfare $\prod_i U_i^{\text{prob}}(p)$.

(2) **Multiplicative Probability Utility:** Primarily used for evaluating distributions via expected utility, forming the basis for Nash Welfare and Proportional Fairness calculations.

$$U_i^{\text{prob}}(X) = \prod_{t=1}^{\ell} r_i^{\text{prob}}(s_{t-1}, a_t) = \prod_{t=1}^{\ell} \pi_i(s_{t-1}, a_t) = P_i(X)$$

This represents the joint probability of sequence X under agent i's policy.

These are related by $U_i^{\log}(X) = \beta \log U_i^{\text{prob}}(X)$, with $U_i^{\text{prob}}(X) > 0$.

3.1 Deterministic vs. Stochastic Policies

We consider two types of policies for generating fair consensus statements in our token-level MDP:

(1) A deterministic policy $\mu(s)$ that gives us a single path $X \in C$. To evaluate deterministic policies, we adopt additive log-utilities $U_i^{\log}(X)$.

(2) A stochastic policy $\pi(a|s)$ that gives us a distribution $p \in \Delta(C)$ over paths (a lottery). When assessing this type of outcome, we consider the expected probability-based utility

$$\mathbb{E}_{X \sim p}[U_i^{\text{prob}}(X)] = \sum_{X \in C} p(X) U_i^{\text{prob}}(X).$$

When it is clear from the context, we adopt the shorthand of $U_i^{\text{prob}}(p)$ to refer to the expected utility of agent i for a given distribution over paths. This expected utility $U_i^{\text{prob}}(p)$ is guaranteed to be non-negative, which is required for downstream fairness measures.

The Fairness of a Deterministic Policy. We assess the fairness of a path $X \in C$ given by a deterministic policy through its egalitarian welfare (EW), drawing from Rawls' maximin principle [20]. This is defined using the additive log-utilities $U_i^{\log}(X)$:

$$\text{EW}^{\log}(X) = \min_{i \in N} U_i^{\log}(X) = \min_{i \in N} \sum_{t=1}^{\ell} \beta \log \pi_i(s_{t-1}, a_t). \quad (1)$$

Maximizing $\text{EW}^{\log}(X)$ means finding the path whose cumulative log-likelihood is highest for the agent who prefers it least. By maximizing the minimum utility, the egalitarian objective promotes broadly acceptable outcomes, which supports the requirement that a consensus statement should be agreeable to all.

The Fairness of a Stochastic Policy. To analyze the fairness of a distribution over paths given by our stochastic policy, we use the non-negative expected utilities $U_i^{\text{prob}}(p)$. Our goal is to find a stochastic policy that gives us a distribution $p \in \Delta(C)$ that is in the ex-ante core [22]. Following Fain et al. [8], we define the ex-ante core as follows.

Definition 1 (Ex-Ante Core). *A distribution $p \in \Delta(C)$ is in the ex-ante core if there is no coalition $S \subseteq N$ and alternative distribution p' such that*

$$\frac{|S|}{|N|} \cdot U_i^{prob}(p') \geq U_i^{prob}(p), \quad \forall i \in S,$$

with strict inequality for at least one agent $i \in S$.

Intuitively, if a distribution over paths is in the ex-ante core, there is no possible coalition of agents that could use their proportional share of probability to create a distribution that has strictly higher expected utility for at least one agent and not less expected utility for all other agents (i.e., a Pareto improvement) [1]. This resistance to coalitional deviation is desirable for consensus statements, as a distribution in the ex-ante core represents an outcome that all parties have implicitly agreed to (since they cannot use their proportional share of utility to do something better).

4 DEFINING A STOCHASTIC POLICY IN THE CORE

Having established the token-level MDP and fairness criteria, we now turn to defining a *stochastic generation policy* π^* that produces a distribution p^* over complete consensus statements C (i.e., a lottery) that is in the ex-ante core.

To do so, we first generate the tree of possible token sequences with branching factor B and maximum length L. We then find the distribution over paths p^* that maximizes Nash welfare (NW). The NW optimal distribution is

$$p^* \in \arg\max_{p \in \Delta(C)} \mathrm{NW}(p), \qquad \mathrm{NW}(p) \coloneqq \prod_{i=1}^{n} U_i^{\mathrm{prob}}(p).$$

We can find p^* by optimizing over the probability simplex $\Delta(C) = \{p \in \mathbb{R}_{\geq 0}^{|C|} : \sum_{X \in C} p(X) = 1\}$. Indexing the leaves as $C = \{X_1, \ldots, X_m\}$ and defining $u_i \in \mathbb{R}_{\geq 0}^m$ by $u_i(j) = U_i^{\mathrm{prob}}(X_j)$, we get that each agent's expected utility under distribution p is

$$U_i^{\mathrm{prob}}(p) = \sum_{j=1}^{m} u_i(j)\, p_j \;=\; u_i^\top p,$$

so the Nash welfare program is

$$p^* = \max_{p \in \Delta(C)} \sum_{i=1}^{n} \log\big(u_i^\top p\big).$$

Note, each term $\log(u_i^\top p)$ is concave (log is concave and increasing; $u_i^\top p$ is affine), and the simplex is convex; hence this is a convex program with polynomial-time algorithms and strong duality under a mild positivity condition (i.e., $u_i^\top p > 0$ for all i) [4].

Moreover, we know that any maximizer of $\prod_i U_i^{\mathrm{prob}}(p)$ lies in the core [1, 7, 8]. So, we can work backwards from this distribution to find a stochastic policy in the core.

Specifically, given the target distribution $p^* \in \Delta(C)$ that maximizes Nash welfare, we derive the stochastic policy π^* that generates this distribution. To define π^*, we introduce some notation. For any state (prefix) s in the token-level MDP:

- Let $C(s) \subseteq C$ be the set of all complete paths (leaves) that pass through state s.
- Let $C(s, a) \subseteq C(s)$ be the subset of paths in $C(s)$ where the next action taken from state s is a. Note that $C(s, a) = C(s\|a)$, where $s\|a$ is the state reached after choosing token a.
- For any subset of leaves $L \subseteq C$, let $P^*(L) = \sum_{X \in L} p^*(X)$ be the total probability mass assigned by the NW optimal distribution p^* to the leaves in L.

Note that $P^*(C(s_0)) = P^*(C) = 1$.

With this, we can define the policy π^* at any given state s.

Definition 2 (Stochastic Policy Induced by p^*). *Let p^* be a distribution over the leaf nodes C. The induced stochastic policy π^* at a non-terminal state s assigns the probability of taking the next action (token) a as:*

$$\pi^*(a|s) = \begin{cases} \frac{P^*(C(s,a))}{P^*(C(s))} & \text{if } P^*(C(s)) > 0 \\ 0 & \text{if } P^*(C(s)) = 0 \end{cases} \tag{2}$$

This represents the conditional probability, according to the NW optimal distribution p^*, of selecting token a next, given that the generation process has reached state s. If state s has zero probability of being reached under p^* (i.e., $P^*(C(s)) = 0$), then the probability of taking any action from s is also zero.

Algorithm 1 summarizes the process of finding a policy in the ex-ante core. We will now briefly turn to the runtime of this algorithm.

Algorithm 1 Compute core-stochastic policy π^*

Require: Leaf set $C = \{X_1, \ldots, X_m\}$, agent utilities $U_i^{\mathrm{prob}}(X_j)$

Build $u_i \in \mathbb{R}_{\geq 0}^m$ with $u_i(j) = U_i^{\mathrm{prob}}(X_j)$
Solve $p^* \in \arg\max_{p \in \Delta(C)} \sum_{i=1}^n \log(u_i^\top p)$
For every state s in the tree, cache $P^*(C(s))$ and, for each enabled action a at s, cache $P^*(C(s, a))$.
if $P^*(C(s)) > 0$ **then**
    $\pi^*(a \mid s) \leftarrow \dfrac{P^*(C(s, a))}{P^*(C(s))}$
else
    $\pi^*(a \mid s) \leftarrow 0$
end if
return π^*

Runtime. We optimize the Nash–welfare program on $\Delta(C)$ (with $m = |C|$ as above) and then induce π^* via the conditional masses $P^*(C(s))$ and $P^*(C(s, a))$. Forming all agent–leaf utilities costs $O(nmL)$. We then minimize $-\sum_i \log(u_i^\top p)$ on the simplex with Frank–Wolfe: each iteration computes $\nabla F(p) = \sum_i u_i/(u_i^\top p)$ in $O(nm)$ time and uses a coordinate linear oracle; the method attains ε-suboptimality in $O(1/\varepsilon)$ iterations [15]. Hence the optimization time is $O(nm/\varepsilon)$. The conditional masses needed for π^* are obtained by a single bottom-up pass over the generation tree, which is linear in the number of leaves when B is fixed, i.e., $\Theta(m)$. Altogether, the total runtime is

$$T_{\mathrm{total}} \;=\; O\big(nmL \;+\; nm/\varepsilon \;+\; m\big),$$

which is polynomial in n, m, and $1/\varepsilon$.

4.1 Properties of the Stochastic Policy

We now establish that executing this policy π^* from the initial state s_0 indeed generates the target distribution p^*.

Theorem 1 (Equivalence of Policy-Induced Distribution and Target Lottery). *Let p_{π^*} be the distribution over C generated by executing the policy π^* (defined in Definition 2) from the initial state s_0. Then $p_{\pi^*} = p^*$.*

The proof is presented in Appendix C.1. This equivalence leads to the desired ex-ante fairness guarantee for the policy π^*.

Corollary 1 (Core Membership of Stochastic Policy). *Let p^* be a distribution over C that maximizes Nash Welfare (and is therefore in the ex-ante core). Let π^* be the stochastic policy derived from p^* according to Definition 2. Then the distribution p_{π^*} generated by executing π^* is in the ex-ante core.*

Proof. By Theorem 1, the distribution generated by policy π^* is $p_{\pi^*} = p^*$. Since p^* was chosen to maximize Nash Welfare over C, it is in the core. Therefore, p_{π^*} is also in the core. □

This simple result confirms that our procedure, which first finds the distribution maximizing Nash Welfare p^* and then executes the derived policy π^*, yields a stochastic policy in the core. And thus, we have an ex-ante fair way to generate consensus statements.

4.2 Empirical Core Test

Here we empirically validate that the NW optimal *policy* π^* is in fact in the ex-ante core, and demonstrate that a uniform policy and a utilitarian optimal policy are not in the ex-ante core.

Environment. For demonstration we use a small B-ary token tree (B=3, depth L=4). Each agent i has a next-token policy

$$\pi_i(a \mid s) \;=\; \mathrm{softmax}_a\big(\rho\, w_i^\top (z + v_{t,a})\big),$$

where z is the running state embedding, $v_{t,a}$ is the token vector, and w_i is the agent vector. The scalar *polarization* $\rho \geq 0$ controls how concentrated preferences are. When $\rho = 0$, $\pi_i(a \mid s) = 1/B$. For two actions a, b,

$$\frac{\pi_i(a \mid s)}{\pi_i(b \mid s)} \;=\; \exp\big(\rho\, w_i^\top (v_{t,a} - v_{t,b})\big),$$

so increasing ρ multiplies odds exponentially and makes agents much more confident about their preferred action.

To initialize the experiment we draw token vectors $\tilde{v}_{t,a} \sim \mathcal{N}(0, I_d)$ and agent vectors $\tilde{w}_i \sim \mathcal{N}(0, I_d)$, set $v_{t,a} = \tilde{v}_{t,a}/\|\tilde{v}_{t,a}\|_2$ and $w_i = \tilde{w}_i/\|\tilde{w}_i\|_2$, and define the state vector at prefix $s = (a_1, \ldots, a_{t-1})$ as $z(s) = \sum_{\tau=1}^{t-1} v_{\tau,a_\tau}$.

Baselines. We compare π^* to a *uniform* policy that induces a uniform distribution over leaves and a *utilitarian* policy that deterministically follows the single path maximizing $\sum_i U_{ij}$.

Blocking test. For any policy π, let p_π be its induced leaf distribution and $u_i(\pi) = U_i^\top p_\pi$. For a coalition S of size r, give it a budget r/n of probability mass and solve a small LP to find the largest factor α such that all $i \in S$ can secure at least $\alpha\, u_i(\pi)$. The *maximum coalition improvement*

$$\alpha^*(\pi) \;=\; \max_{S \neq \emptyset} \; \max_{p' : 1^\top p' = r/n,\, p' \geq 0} \; \min_{i \in S} \frac{U_i^\top p'}{u_i(\pi)}$$

The value of $\alpha^*(\pi)$ is > 1 exactly when a coalition can block π. Figure 2 plots $\alpha^*(\pi)$ versus polarization ρ (log y-axis).

Takeaways. Figure 2 shows the result. As ρ increases, agents' own policies become more concentrated. The Nash-welfare *policy* stays at $\alpha^*(\pi^*) = 1$ across all ρ, consistent with the theoretical result. The utilitarian policy becomes highly blockable, and the uniform policy becomes steadily easier to block.

4.3 Computational Tractability via Token Chunking

The set of complete statements C grows as $|C| \approx B^{L_{\max}}$ for branching factor B and maximum length $L_{\max}$. Enumerating leaves and solving for p^* over C could be infeasible for long sequences.

We therefore restrict attention to a coarser action space obtained by grouping tokens into macro-actions.

Definition 3 (Token chunking). *A chunking scheme $\mathcal{K}$ partitions positions into contiguous blocks $\{k_1, \ldots, k_m\}$, where each k_j contains one or more tokens and the concatenation of chosen blocks forms a complete statement. Decoding now selects entire blocks. The corresponding feasible set of leaves is $C_{\mathcal{K}} \subseteq C$.*

With fixed chunk size c, the effective depth drops from L to $\lceil L/c \rceil$. Running Algorithm 1 on the chunked tree produces a lottery that lies in the ex-ante core *with respect to* $C_{\mathcal{K}}$; all coalitions are evaluated against this feasible set. This interpretation mirrors real elections: not every eligible candidate runs, yet we judge the outcome by comparing the candidates who did run. Here, chunking plays the role of eligibility: it narrows the set of feasible statements, and fairness is defined relative to that set.

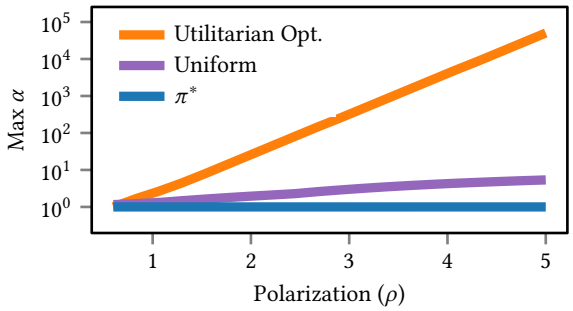

Figure 2: Maximum coalition improvement vs. polarization.

However, this restriction does have a consequence. Because $C_{\mathcal{K}} \subseteq C$, the optimal Nash welfare over $C_{\mathcal{K}}$ cannot exceed that over C. Moreover, no uniform multiplicative guarantee is possible without further structure:

Theorem 2 (No constant-factor approximation under chunking). *For any constant $R > 1$, there exists a two-agent instance, a finite set of leaves C, and a chunked subset $K \subset C$ such that, writing*

$$\mathrm{NW}_Y(p) \;=\; \prod_{i=1}^{2} \Big(\sum_{x \in Y} u_i(x)\, p(x) \Big) \quad \text{for } Y \subseteq C,\; p \in \Delta(Y),$$

and letting $p_Y^* \in \arg\max_{p \in \Delta(Y)} \mathrm{NW}_Y(p)$, we have

$$\frac{\mathrm{NW}_C(p_C^*)}{\mathrm{NW}_K(p_K^*)} \;>\; R.$$

Hence the approximation ratio of the chunked solution, measured against the unchunked optimum, is unbounded.

The proof appears in the Appendix. The theorem states a worst case that arises when chunking prunes precisely those statements that both agents value highly.

Observe that we can think about this as an anytime algorithm: enlarging $C_{\mathcal{K}}$ can only improve the Nash welfare, and the ex-ante core guarantee continues to hold relative to the current feasible set.

5 GENERATING A SINGLE STATEMENT

Although our stochastic policy π^* achieves a highly desirable ex-ante fairness guarantee, many practical applications require selecting a single consensus statement. In this case, our objective shifts from finding a fair distribution to identifying the single path (i.e., statement) that represents consensus.

5.1 Finding the Egalitarian Path

Given the token tree with leaf nodes C, we aim to find a deterministic policy μ^* that produces a single path $X^* \in C$ that maximizes egalitarian welfare as defined in Equation 1. Due to the size of the

token tree, exhaustive search for X^* may be intractable. We propose approximate algorithms to find high-quality paths, including finite-lookahead search and beam search, which are detailed below.

Finite Lookahead Search. The finite lookahead algorithm operates with a rolling horizon. At each step t, it explores all possible paths P of length up to d originating from s_t. For each such path P, the algorithm evaluates the egalitarian welfare of the sequence formed by concatenating the path generated so far (X_{prefix}) with P. It then chooses the first action a^* of the path P^* that maximizes this lookahead evaluation, transitions to state $s_{t+1} = T(s_t, a^*)$, and repeats the process. This d-step lookahead can mitigate the potential for hedging inherent in greedy search. When no single immediate token is agreeable (i.e., results in high egalitarian welfare), a greedy method might select less informative tokens that avoid commitment. In contrast, a lookahead can identify longer sequences that, despite potentially controversial initial steps, lead to states with higher overall welfare, perhaps by expressing a concept with suitable qualifications. The algorithm is shown in Appendix D.

Beam Search. Beam search is a heuristic search algorithm that balances greedy search and exhaustive exploration, and has been effective in sequence generation tasks like machine translation and text generation [16, 18]. Instead of pursuing only the single best option (greedy search) or all options (exhaustive search), beam search maintains a fixed number of the most promising partial paths (hypotheses), w (the beam width), at each depth t. At each step, it expands paths in the beam by generating potential successor tokens. These candidates are then evaluated using the egalitarian welfare objective function, and only the top w scoring paths are retained for the next step. The algorithm returns the highest-scoring complete path found within the beam at the maximum length or upon reaching a terminal state. The algorithm is shown in Appendix D.

6 EXPERIMENTS

We conduct experiments with three main objectives. First, we test whether agent policies derived from prompting language models show meaningful correlation with human preferences. Second, we examine whether these prompted policies can perform credit assignment to generate meaningful token-level rewards (a necessary prerequisite for our search algorithms). Third, we evaluate how well our finite lookahead and beam search methods perform in generating single consensus statements when compared to baseline approaches.

6.1 Opinion-length scaling experiment

We test whether a chat LLM's conditional likelihood of a policy statement, given a participant's written opinion, predicts that participant's 1–5 Likert rating, and whether supplying more of the participant's opinion improves prediction.

Data. We use the abortion dataset from Fish et al. [10]. This dataset contains data from 42 human participants who each provided their opinion on abortion in natural language. Each participant also rated five other candidate statements about abortion using a 1-5 Likert scale where higher values indicate higher agreement.

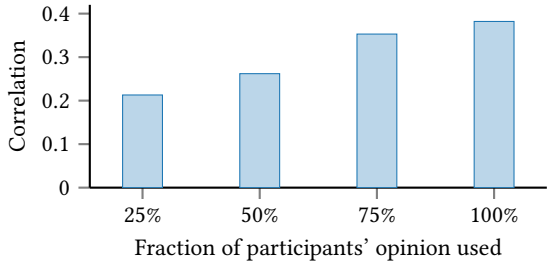

Figure 3: Spearman's correlation between model likelihood and participants' Likert ratings vs. the fraction of the user's opinion provided.

Method. We evaluate how well candidate statements capture participants' opinions using language model scoring. For each participant u and statement j, we construct a paraphrase validation task using Meta-Llama-3.1-8B-Instruct-Turbo with temperature 1.0, computing log-probabilities without generation.

We structure the prompt as follows: the system instruction establishes the paraphrasing task, the user message provides the issue topic and the participant's original opinion, and we place the candidate statement as a pre-filled assistant response:

```
[System] You paraphrase people's views.
[User] Topic: {issue}
       Original opinion: {opinion}
[Assistant] Paraphrase: {statement}
```

Given the assistant response following the "`Paraphrase:`" tokenized as $a_1, \ldots, a_M$, we compute the length-normalized log-probability score:

$$s_{u,j} = \frac{1}{M} \sum_{k=1}^{M} \log p_\theta(a_k \mid \text{context}, a_{1:k-1})$$

Intuitively, this score measures how well the statement represents the participant's opinion. For validation, we compute Spearman's correlation between each participant's five model-assigned scores $\{s_{u,j}\}_{j=1}^{5}$ and their corresponding Likert ratings for the same statements.

Length manipulation. We repeat the procedure after truncating each participant's opinion to the last fraction $f \in \{1.00, 0.75, 0.50, 0.25\}$ of its words.

Results. The correlation between the average likelihood of the statement according to the user prompted policy and the user's rating increases with the fraction of the user's opinion that is conditioned on (Fig. 3). Thus, conditioning on more of the opinion yields better predictions of human ratings. We note that with the users' full opinions the correlation is still somewhat low, but the positive trend observed suggests that eliciting more informative statements could lead to a more accurate signal.

6.2 Evaluating Credit Assignment

Setup. Our framework defines token-level rewards as $r_i(s, a) = \beta \log \pi_i(a \mid s)$, where each policy π_i is obtained by prompting a base LLM with agent-specific information. Reward-guided search is

Table 1: Credit assignment results for `Llama 3.1 8B Instruction-Tuned`. Darker green indicates larger Z-score. Z-score column is for altered tokens. Alterations are represented by "<misaligned>/<aligned>".

Reference policy prompt	User policy prompt	Sequence	Z-Score
User food profile: empty	User food profile: vegetarian	I am having chicken/tofu enchiladas tonight. Then I am going to meet up with some friends.	2.69
User location profile: empty	User location profile: lives in a cold climate	I 'm going to the beach/mountains this weekend to surf/ski . I need to buy some new clothes .	1.78, 3.31
User time profile: empty	User time profile: morning	I am about to eat some food . I am going to have spaghetti/pancakes . I will use my phone to order it .	4.26
User opinion: empty	User opinion: Favors stricter gun control laws.	Implement ing background checks that are less/more strict for gun purchases is essential . Also , my favorite color is orange .	2.09

effective when policies exhibit *localized credit assignment*: changes in $\pi_i(a \mid s)$ concentrate on tokens tied to the prompt information. Rafailov et al. [18] observed this behavior in DPO-trained models; here we test it for policies induced by prompting instruction-tuned models.

Test design. We use Llama 3.1 8B Instruction-Tuned. For each test, we compare token log-probabilities under a **user policy prompt** (e.g., "User time profile: morning") and a **reference policy prompt** (e.g., "User time profile: empty"). We evaluate two nearly identical sequences: X_1 contains a concept that conflicts with the user profile (e.g., "spaghetti" for a "morning" profile) and X_2 replaces it with an aligned concept (e.g., "pancakes").

Metric. For each token a_j with prefix s, we compute the log-likelihood difference between the user and reference policies,

$$\Delta L(a_j \mid s) = \log \pi_U(a_j \mid s) - \log \pi_R(a_j \mid s).$$

We then measure the change in this difference when switching from X_1 to X_2,

$$D_j = \left| \Delta L_{X_1}(a_j \mid s) - \Delta L_{X_2}(a_j \mid s) \right|.$$

A large D_j indicates that the user profile alters the model's preference for token a_j specifically when the conflicting concept is swapped for an aligned concept. To compare across positions, we convert $\{D_j\}$ to Z-scores using the mean and standard deviation over all tokens in the sequence.

Findings. The largest Z-scores occur at the tokens that differ between X_1 and X_2 (Table 1). With a "morning" time profile, the "spaghetti" versus "pancakes" position shows the highest shift. Additional examples with Llama and `Gemma 2 9B Instruction-Tuned` in Appendix E.1 show the same pattern.

Implication. System prompting induces localized credit assignment in instruction-tuned models. This supports using $r_i(s, a) = \beta \log \pi_i(a \mid s)$ from prompted policies as a targeted signal for token-level search.

6.3 Consensus Generation

We evaluated different approaches for generating a single consensus statement by comparing our proposed search algorithms against several baselines. The primary goal was to assess how well each method optimizes egalitarian welfare (EW), measured by a perplexity-based metric reflecting worst-case agent alignment. More detailed consensus generation experiments, with `Gemma 2 9b Instruction-Tuned`, an LLM-judge metric, and with more agents are presented in subsection E.4 of Appendix E.

Scenarios: We used scenarios from the Habermas Machine dataset [24]. To obtain distinct settings, scenario descriptions were embedded using `BAAI/bge-large-en-v1.5` [25] and clustered via k-means ($k = 3$). Representative scenarios were selected from each cluster (Scenarios 1, 2, and 3). The issues for these scenarios are in the captions of Tables 4, 5, and 6.

Agents and Policies: For each scenario, agent opinions were taken from the dataset. Agent policies π_i were instantiated by prompting `Llama 3.1 8B Instruct` [12] with the issue and agent i's opinion, instructing it to generate text aligned with that viewpoint (prompt shown in Figure 7 in Appendix F). The resulting likelihoods $\pi_i(s, a)$ represent agent i's preferences. Agent opinions are detailed in Tables 4, 5, 6 in subsection E.3 of Appendix E.

Base Generation Model: Consensus statements were generated using `Llama 3.1 8B Instruct`, prompted with the issue and all agent opinions (prompt shown in Figure 6 in Appendix F).

Evaluation Metric - Egalitarian Perplexity: To capture alignment with the least satisfied agent for a consensus statement X, we define Egalitarian Perplexity (EPPL). For each agent i, their specific perplexity $PPL_i(X)$ is found by prompting `Llama 3.1 8B Instruct` with the issue and agent i's opinion to generate a statement perfectly reflecting that opinion. The average log-likelihood of the actual consensus statement $X = (a_1, \ldots, a_L)$ conditioned on this agent-specific prompt is:

$$\bar{L}_i(X) = \frac{1}{L} \sum_{t=1}^{L} \log \pi_i(s_{t-1}, a_t | \text{prompt}_i).$$

The agent-specific perplexity is $PPL_i(X) = \exp(-\bar{L}_i(X))$. The final Egalitarian Perplexity for X is $EPPL(X) = \max_{i \in N} PPL_i(X)$. Lower EPPL indicates better egalitarian welfare.

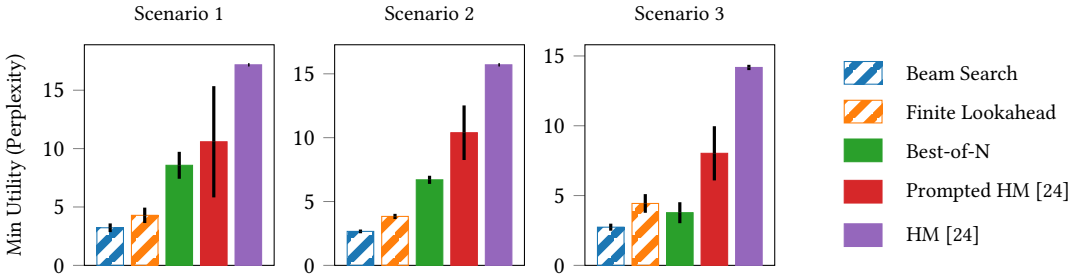

Figure 4: Per-scenario egalitarian welfare (perplexity). Lower values indicate better minimum agent utility. Striped bars indicate that the method uses search over the token-level MDP. Numerical results are reported in Table 3 in Appendix E.

Seeds: We report the mean and standard deviation of EPPL over 3 seeds per method and scenario.

Methods Compared: We compared our proposed algorithms, **Finite Lookahead** (Algorithm 2 in Appendix D, depth $d = 4$, branching $B = 2$), and **Beam Search** (Algorithm 3 in Appendix D, width $w = 4$, pruning based on partial EPPL), against three baselines: **Best-of-N** (selecting the best of $N = 4$ samples[1] from the base model by EPPL); a **Prompted Habermas Machine**[2] (1 critique round, 4 candidates[3], critiques from base model conditioned on agent opinions); and the original **Habermas Machine** (HM) [24] consensus (generated by a fine-tuned Chinchilla-70B).

6.3.1 Results. Figure 4 summarizes EPPL performance (lower is better). **Beam Search consistently achieved the lowest EPPL** (average: 2.87), indicating high alignment with the least satisfied agent. **Finite Lookahead also performed well** (average EPPL: 4.18), outperforming baseline methods. Both search methods surpassed **Best-of-N** (6.35) and the **Prompted Habermas Machine** (9.67). The **Habermas Machine** baseline had the highest EPPL (15.69), possibly because its statement was generated by a different model (Chinchilla 70B).

The results suggest that token-level search guided by EPPL, as in Beam Search and Finite Lookahead, effectively generates consensus statements with better minimum agent alignment compared to sampling or iterative refinement. Consensus statements for the first seed are in Tables 4, 5, and 6 in Appendix E. The strong empirical performance of methods operating on the token-level MDP complements the fact that these methods are also more amenable to theoretical analysis and fairness guarantees.

7 DISCUSSION

This work introduced a framework for generating consensus statements by modeling the process as a multi-objective, token-level MDP with rewards derived from agent-specific language model policies. Our aim was to connect LLM-based text generation with the formal fairness guarantees of social choice theory via this MDP.

Our theoretical contributions for stochastic outcomes (lotteries over statements) focused on the core. By maximizing Nash Welfare over expected probability-based utilities, we identified an optimal lottery p^* that induces a stochastic generation policy π^* (Definition 2) inheriting the ex-ante core property (Corollary 1). Chunking was introduced as a heuristic to manage the search space. For deterministic outcomes (single statements), we focused on maximizing egalitarian welfare (EW), proposing finite lookahead and beam search as approximation algorithms.

Empirical results validated several aspects of our framework. We found that token likelihood from prompted policies meaningfully correlate with human preferences. Credit assignment experiments (subsection 6.2) confirmed that prompting LLMs with agent profiles creates policies that not only correlate with human preferences, but that also correctly assign credit to tokens. Consensus generation experiments (subsection 6.3), using Egalitarian Perplexity (EPPL) to measure EW, showed that beam search and finite lookahead, guided by the EW objective, outperformed baselines like Best-of-N and an adapted Habermas Machine. Beam search yielded the lowest EPPL.

Overall, formulating consensus generation as a search problem within a token-level MDP, guided by explicit social choice objectives like EW, is a promising direction. However, there are some important questions that future work should address. First, finding methods to train more faithful personalized policies for each agent is an important direction as it is upstream of many important challenges [2]. Second, finding a way to approximate the core on the unchunked space without looking at the whole tree is an important step to work towards. Previous work has looked at approximating the core [7, 11], but neither method directly applies to our setting. And third, future work should focus on theoretical guarantees for the single statement case, which we did not obtain.

Lastly, we note that until our methods are better understood, the outputs of our algorithm should be treated as artifacts for collective sense-making instead of binding decisions, as suggested by Revel and Pénigaud [21]. For example, instead of treating the output as a decision, the output could be treated as another input to the discussion that participants of the collective decision could reflect on. Optimistically, one would hope that these consensus statements could identify previously unknown points of agreement or solutions that no one had previously thought of that are in fact highly agreeable. In sum, this work contributes theoretical foundations and practical algorithms for incorporating social choice principles into generative AI for collective decision-making and sense-making.

[1] $N = 4$ was chosen to align with the Prompted Habermas Machine.
[2] As implemented in https://github.com/google-deepmind/habermas_machine
[3] We chose four candidates to align with the default parameters in the Prompted Habermas Machine example in the Habermas Machine GitHub repository.

REFERENCES

[1] Haris Aziz, Anna Bogomolnaia, and Hervé Moulin. 2019. Fair mixing: The case of dichotomous preferences. In *Proceedings of the 2019 ACM Conference on Economics and Computation*. 753–781.

[2] Carter Blair, Kate Larson, and Edith Law. 2025. Reflective Verbal Reward Design for Pluralistic Alignment. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, James Kwok (Ed.). International Joint Conferences on Artificial Intelligence Organization, 10271–10279. https://doi.org/10.24963/ijcai.2025/1141 Human-Centred AI.

[3] Niclas Boehmer, Sara Fish, and Ariel D. Procaccia. 2025. Generative Social Choice: The Next Generation. In *Forty-second International Conference on Machine Learning*. https://openreview.net/forum?id=E1E6T7KHlR

[4] Stephen P Boyd and Lieven Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press.

[5] Souradip Chakraborty, Sujay Bhatt, Udari Madhushani Sehwag, Soumya Suvra Ghosal, Jiahao Qiu, Mengdi Wang, Dinesh Manocha, Furong Huang, Alec Koppel, and Sumitra Ganesh. 2025. Collab: Controlled Decoding using Mixture of Agents for LLM Alignment. In *The Thirteenth International Conference on Learning Representations*. https://openreview.net/forum?id=7ohlQUbTpp

[6] Vincent Conitzer, Rupert Freeman, and Nisarg Shah. 2017. Fair public decision making. In *Proceedings of the 2017 ACM Conference on Economics and Computation*. 629–646.

[7] Soroush Ebadian, Anson Kahng, Dominik Peters, and Nisarg Shah. 2024. Optimized distortion and proportional fairness in voting. *ACM Transactions on Economics and Computation* 12, 1 (2024), 1–39.

[8] Brandon Fain, Kamesh Munagala, and Nisarg Shah. 2018. Fair allocation of indivisible public goods. In *Proceedings of the 2018 ACM Conference on Economics and Computation*. 575–592.

[9] Shangbin Feng, Chan Young Park, Yuhan Liu, and Yulia Tsvetkov. 2023. From Pretraining Data to Language Models to Downstream Tasks: Tracking the Trails of Political Biases Leading to Unfair NLP Models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 11737–11762. https://doi.org/10.18653/v1/2023.acl-long.656

[10] Sara Fish, Paul Gölz, David C. Parkes, Ariel D. Procaccia, Gili Rusak, Itai Shapira, and Manuel Wüthrich. 2024. Generative Social Choice. In *Proceedings of the 25th ACM Conference on Economics and Computation* (New Haven, CT, USA) *(EC '24)*. Association for Computing Machinery, New York, NY, USA, 985. https://doi.org/10.1145/3670865.3673547

[11] Ian Gemp, Marc Lanctot, Luke Marris, Yiran Mao, Edgar Duéñez-Guzmán, Sarah Perrin, Andras Gyorgy, Romuald Elie, Georgios Piliouras, Michael Kaisers, et al. 2024. Approximating the Core via Iterative Coalition Sampling. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*. 669–678.

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

[13] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948* (2025).

[14] Seungwook Han, Idan Shenfeld, Akash Srivastava, Yoon Kim, and Pulkit Agrawal. 2024. Value augmented sampling for language model alignment and personalization. *arXiv preprint arXiv:2405.06639* (2024).

[15] Martin Jaggi. 2013. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *International conference on machine learning*. PMLR, 427–435.

[16] Daniel Jurafsky and James H. Martin. 2026. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, with Language Models* (3rd ed.). https://web.stanford.edu/~jurafsky/slp3/ Online manuscript released January 6, 2026.

[17] Jiacheng Liu, Andrew Cohen, Ramakanth Pasunuru, Yejin Choi, Hannaneh Hajishirzi, and Asli Celikyilmaz. 2024. Don't throw away your value model! Generating more preferable text with Value-Guided Monte-Carlo Tree Search decoding. In *First Conference on Language Modeling*. https://openreview.net/forum?id=kh9Zt2Ldmn

[18] Rafael Rafailov, Joey Hejna, Ryan Park, and Chelsea Finn. 2024. From r to Q^*: Your Language Model is Secretly a Q-Function. In *First Conference on Language Modeling*. https://openreview.net/forum?id=kEVcNxtqXk

[19] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Thirty-seventh Conference on Neural Information Processing Systems*. https://openreview.net/forum?id=HPuSIXJaa9

[20] John Rawls. 1971. *A Theory of Justice: Original Edition*. Harvard University Press. http://www.jstor.org/stable/j.ctvjf9z6v

[21] Manon Revel and Théophile Pénigaud. 2025. AI-Enhanced Deliberative Democracy and the Future of the Collective Will. (July 2025). https://doi.org/10.48550/arXiv.2503.05830 working paper or preprint.

[22] Lloyd S Shapley. 1971. Cores of convex games. *International Journal of Game Theory* 1 (1971), 11–26.

[23] Ruizhe Shi, Yifang Chen, Yushi Hu, Alisa Liu, Hanna Hajishirzi, Noah A Smith, and Simon S Du. 2024. Decoding-time language model alignment with multiple objectives. *Advances in Neural Information Processing Systems* 37 (2024), 48875–48920.

[24] Michael Henry Tessler, Michiel A. Bakker, Daniel Jarrett, Hannah Sheahan, Martin J. Chadwick, Raphael Koster, Georgina Evans, Lucy Campbell-Gillingham, Tantum Collins, David C. Parkes, Matthew Botvinick, and Christopher Summerfield. 2024. AI can help humans find common ground in democratic deliberation. *Science* 386, 6719 (2024), eadq2852. https://doi.org/10.1126/science.adq2852

[25] Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-Pack: Packaged Resources To Advance General Chinese Embedding. arXiv:2309.07597 [cs.CL]