LLMs in GT, Racial Bias, Diplomacy and piKL

Armaan Khetarpaul

1 LLMs in GT

1.1 Can Large Language Models Serve as Rational Players in Game Theory? A Systematic Analysis

1.1.1 Introduction

Three games (dictator game, RPS, and ring-network game) are used to analyse behaviour of LLMs

1.1.2 Experiments

Can LLMs build a Clear Desire?

The Dictator Game: Dictator has two allocation options, recipient must accept.

$$\pi^*(a|\mathcal{I}) = \underset{a\in\{X,Y\}}{argmax}\{D(X,\omega_{\mathcal{I}}), D(X,\omega_{\mathcal{I}})\}$$

Four preference types: *Equality, Common Interest, Selfishness, and Altruism.* In game theory, SI and EQ are the most common preferences, followed by CI, while AL hardly ever occurs.
Preferences are assigned to LLM, plays 3 games, repeated 10 times.

LLM	Pref.	Option			
		EQ	CI	SI	AL
GPT-3	EQ	-	1.0	1.0	1.0
	CI	0.4	-	0.3	0.5
	SI	1.0	1.0	-	1.0
	AL	0.0	0.0	0.1	-
GPT-3.5	EQ	-	1.0	1.0	1.0
	CI	1.0	-	0.9	1.0
	SI	1.0	1.0	-	1.0
	AL	1.0	0.6	0.8	-
GPT-4	EQ	-	1.0	1.0	1.0
	CI	1.0	-	1.0	0.9
	SI	1.0	1.0	-	1.0
	AL	1.0	1.0	1.0	-

When assigned common preferences (EQ and SI), all three LLMs made preference-consistent choices in all experiments, demonstrating the basic ability of LLMs to build clear desires from textual prompts. However, LLMs performed poorly when given uncommon preferences (CI and AL). Mathematical ability of LLMs assigned different preferences would be significantly different (GPT-3 and GPT-3.5). **LLMs have the basic ability to build clear desires based on textual prompts, but struggle to build desires from uncommon preferences.**

Can LLMs Refine Belief?

Rock-Paper-Scissors: LLMs play RPS. Optimal strategy for round t:

$$\pi^*(a_m^t|\mathcal{I}) = \underset{a_m^t \in \mathcal{A}}{argmax} \mathbb{E}_{a_o^t \sim p(\Omega_{\{a_o^{<t}, a_m^{<t}\}})} \left[D(a_o^t, a_m^t) \right]$$

Strategy	Name	Description
$a_o^t = C$	constant	remain constant
$a_o^t = f(a_o^{<t})$	loop-2 loop-3	loop between two actions loop among three actions
$a_o^t = f(a_m^{<t})$	copy counter	copy opponent's previous action counter opponent's previous action
$a_o^t \sim p(\mathcal{P})$	sample	sample in preference probability

Table 2: Summary of the opponent's strategy in R-S-P.

4 opponent patterns. For constant, we perform 3 experiments (R, P and S). For loop 2, we perform 3 experiments (R-P, P-S, S-R). For loop 3 we perform one experiment (R-P-S). For markov, we perform two experiments copy and counter. Finally we perform experiment with preference distributions as (0.7,0.15,0.15) for RPS. **Analysis:**

- In the basic pattern (constant), GPT-3 performed close to random guessing, suggesting that GPT-3 lacked the basic ability to refine belief. In contrast, GPT3.5's average payoff was significantly higher than random guessing and continued to rise; GPT-4 consistently took correct actions after approximately 3 rounds.
- For loop 2 and loop 3, GPT-3 and 3.5 struggled but GPT-4 performed well.
- For markov the situation was not ideal, so all the LLMs struggled.
- For the preference distribution, LLMs didn't recognize the pattern, and played randomly.

Currently, the ability of LLMs to refine belief is still immature and cannot refine belief from many specific patterns (even if simple). However, GPT4 still had good results.

Can LLMs Take Optimal Actions?

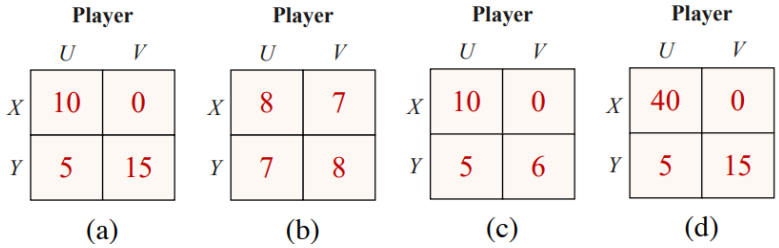
Ring-Network Game: It's a 2 player game with both players having a payoff matrix. The characteristic of the ring-network game is that players' optimal action is determined sequentially by the other players' optimal actions.

$$\pi^*(a_m|\mathcal{I}) = \underset{a_m \in \{U,V\}}{argmax} \left[p(a_o|M) \cdot D_m(a_m|a_o, M) \right]$$

(my actions = $\{U, V\}$ opponent actions $\{X, Y\}$) M is payoff bimatrix. 3 forms of beliefs:

- Implicit: $LLM(a_m|M)$
- Given: $LLM(a_m|a_o, M)$
- Explicit: Analyse and refine $LLM(a_o|M)$. And then play $LLM(a_m|a_o, M)$

We keep opponent's payoff matrix constant.



We set up different player' payoff matrices, to adjust the difficulty of taking the optimal action: (a) is the original setup; (b) reduces the difference in payoffs while keeping the expected payoffs to $a_m \in \{U, V\}$ constant; (c) increases the expected payoff for the incorrect action $a_m = U$; and (d) decreases the expected payoff for the correct action $a_m = V$.
GPT-3 performed worse in all cases. All LLMs performed bad in Implicit belief. For Explicit belief, even though the LLMs could refine the beliefs (95%), they were unable to make optimal actions on refined beliefs (70%). But in the case of Given belief, GPT-4 was able to perform the optimal action consistently, and GPT-3.5 was able to perform the optimal action in 80% of the cases.
Two types of mistakes were made:

- Belief is overlooked: LLMs are confused by the game information and thus overlook the refined belief to take the optimal action in the subsequent dialogue.
- Belief is modified: LLMs lack confidence in the refined belief and thus modify the refined belief to take the optimal action in the subsequent dialogue.

We consider that LLMs do not have the ability to autonomously follow human behavior in the game process. As a result, it is necessary to explicitly decouple human behavior for LLMs in game theory.

1.1.3 Results

With the dictator game, we find that LLMs have the basic ability to build a clear desire. However, when assigned uncommon preferences, LLMs often suffer from decreased mathematical ability and inability to understand preferences. With Rock-Paper-Scissors, we observe that LLMs cannot refine belief from many simple patterns, which makes us pessimistic about LLMs playing games that require refining complex beliefs. Nonetheless, GPT-4 exhibits astonishingly human-like performance in certain patterns, able to become increasingly confident of refined belief as the game information increases. With the ring-network game, we consider that LLMs cannot autonomously follow the player's behavior. Explicitly decomposing the behavior in the game process can improve the ability of LLMs to take optimal actions, but the phenomenon of overlooking / modifying refined belief remains unavoidable in LLMs. In summary, our research systematicall

1.2 States as Strings as Strategies: Steering Language Models with Game-Theoretic Solvers

1.2.1 Introduction

We consider extensive-form games (multiple players, dependent on history, etc.). We represent dialogues as a game by using states and assume perfect recall for each player.

1.2.2 Language as Strategy

We construct a version of Policy-Space Response-Oracles (PSRO) where an approximate best response can be generated by sampling new random prompt strings, evaluating them against the current equilibrium, and then returning the one with highest expected payoff. Some algorithms for this are:

Algorithm 1 Shotgun Approximate Best Response

Input: Focal agent i
Input: Current joint policy π
Input: Number of shotgun candidates k
  C is the current action set with their scores under π
  for $t = 1 \leq k$ **do**
    Prompt LLM to generate new candidate $c_t \cap C = \emptyset$
    Evaluate candidate c_t against policy π_{-i} to give score s_t
    $C = C \cup \{(c_t, s_t)\}$
  end for
Output: c_t with max s_t

Algorithm 2 Approximate Better Response

Input: Focal agent i and its score s^* under π
Input: Current joint policy π
  while $s \leq s^*$ **do**
    Prompt LLM to generate new candidate c
    Evaluate candidate c against policy π_{-i} to give score s
  end while
Output: c

Algorithm 3 Trajectory-Aware Approximate Best Response

Input: Focal agent i
Input: Current joint policy π
Input: Number of candidates k
  C is the current action set with their scores under π
  Order C by their scores in ascending order
  Prompt LLM to generate k new candidates in order of ascending score given ranked C
  Evaluate new candidates against policy π_{-i} to give scores
Output: c_t with max s_t

Algorithm 4 Categorical Approximate Best Response

Input: Focal agent i
Input: Current joint policy π
Input: Number of candidates per category k
Input: Number of category candidates k'
  C is the current set of action categories with their (Nash) average scores under π
  Order C by their scores in ascending order
  Prompt LLM to generate k' new candidate categories in order of ascending score given ranked C
  Prompt LLM to generate k candidates for each new action category
  Evaluate new candidates against policy π_{-i} to give scores
Output: Category with highest average score

Algorithm 5 Prompt-Space Response-Oracles

Input: C where C_i is the initial prompt action set (singleton) for player i
Input: h containing hyperparameters for approximate best response operator BR
  Compute expected payoff tensor P over joint action(s) C
  π is uniform meta-strategy profile over C
  `incomplete = True`
  while `incomplete` **do**
    for player $i \in [N]$ **do**
      $c_i \leftarrow \text{BR}(i, \pi, h)$, e.g., Algorithms (1-4)
    end for
    if $c_i \in C_i \ \forall i \in [N]$ **then**
      `incomplete = False`
    else
      $C_i \leftarrow C_i \cup c_i \ \forall i \in [N]$
      Compute expected payoff tensor P over joint actions C
      $\pi \leftarrow$ meta-strategy w.r.t. P
    end if
  end while
Output: (π, C, P)

OpenSpiel is a game theory engine with a large community of contributors. OpenSpiel provides several game-theoretic solvers already; this allows someone without experience in computational game theory to focus on the modelling of their dialogue game rather than how to design and implement a game solver

1.2.3 Games

Soome games using PSRO (Algorithm 5) using a shotgun approach for a best response operator (Algorithm 1) are:

- **Scheduling a Meeting** Players attempt to schedule a meeting through a multi-turn negotiation. Each player begins with a set of allowable days of the week, i.e., days in which they are available to meet: ""Sunday", ""Monday", . . . , ""Saturday". They also have non-negative valuations over each day of the week (distinct from the allowable days). Both of these pieces of information are private to the players. Players can choose to reveal this information if they wish. Naturally, their actions here are the days of the week on which they propose to meet. The game rewards players according to how much they value the agreed upon day, and receive zero reward if no agreement is made.

- **Trading Fruit** Each player begins with a private endowment of fruit (i.e., a fruit basket) as well as private valuations over types of fruit. Players are rewarded by the difference in value between their basket after the trade and that before the trade. In addition, players can choose to adjust the "tone" of their negotiations.

- **Public Debate** We present two LLMs with an argument topic; one is tasked with arguing for the statement, one against. Each player's performance in the debate is scored between 0 and 1. We considered twenty different debate topics.

And we also explore an off-the-OpenSpiel-shelf counterfactual regret minimization (CFR) approach which we extend to unseen domains via imitation learning.

1.2.4 Experiments and Results

LLM base is PaLM 2 S. Evaluation requires assessing both our game-theoretic model as well as the performance improvement provided by game-theoretic solvers.

Evaluation as Game Models

Given an LLM generated a message m conditioned on a prompt formatted with an action a, we would like to determine if a is actually the most likely action conditioned on m using a held-out model p, i.e., $a = \underset{z \in \mathcal{A}}{argmax} p(z|m)$. We use the same LLM as our held-out model p.

Evaluation as Reward Models

It is difficult to extract from the natural language conversation the exact deal (or no deal) that is agreed upon with hand-coded parsing. One failure mode we noticed was that the LLM-based reward model would assume a trade agreement had been reached (and calculate the corresponding trade value) even when the final sent message was a counter proposal. As with many parts of this LLM-based game, any improvements in the language models, prompting, or dialogue flow can lead to improvements in the ability of the game to represent realistic interactions.

Evaluation as Game-Theoretic Solvers

2 algorithmic approaches:

- **Counterfactual Regret Minimization** We simply call OpenSpiel's built-in counterfactual regret minimization (CFR) solver on our open sourced chat_game. We do this for many games, procedurally generated for both the debate and meeting scheduling (with days-of-the-week as actions) domains. On average, we find that CFR returns an improved strategy over letting the LLM choose its responses without in-context direction ("any" is passed to the LLM as an action in this case).

Domain	# of Samples	Min/Max Payoff	NashConv	CFR Gain
Debate	328	0/1	0.024	0.106
Schedule Meeting (DOW)	67	0/20	0.417	1.596

- **Prompt-Space Response Oracles** Prompt-Space Response-Oracles (Algorithm 5) alternates between solving for an equilibrium of the game and then approximating a best response to this equilibrium.

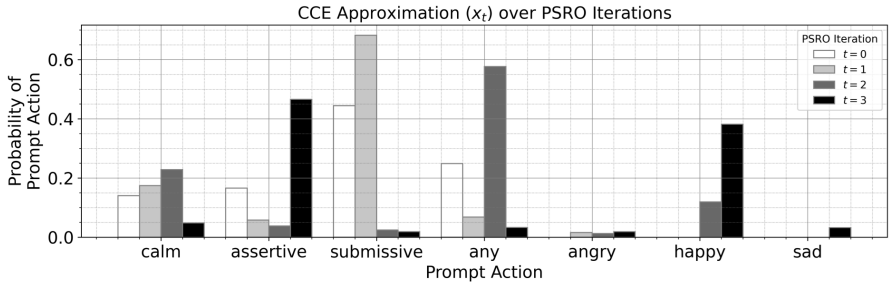


Figure 3 | Prompt-Space Response-Oracles (Algorithm 5) demonstration on meeting scheduling domain where actions are tones. Which day of the week to propose is left up to the LLM.

We solved for this equilibrium using replicator dynamics. This game is general-sum, in which case, replicator dynamics only guarantees convergence of the time average policy to a coarse-correlated equilibrium (CCE).

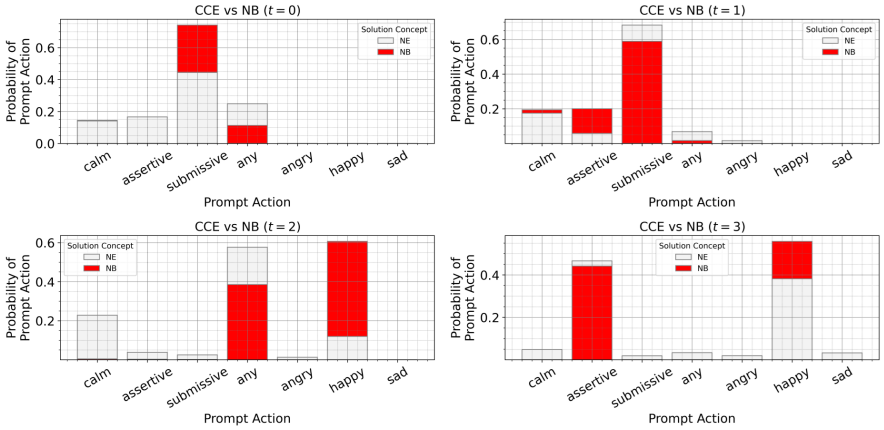


Figure 4 | The same PSRO run as Figure 3 but additionally reporting the Nash bargaining solution in red at each iteration for the meeting scheduling domain.

"Submissive" is initially the most probable action at equilibrium, but equilibria are far from the only solution concepts proposed and studied in game theory. The Nash bargaining solution is the unique solution to a two-person bargaining problem that satisfies the axioms of scale invariance, symmetry, efficiency, and independence of irrelevant alternatives. In this case, NB and CCE roughly agree in terms of their mixed strategies on the meeting scheduling domain.

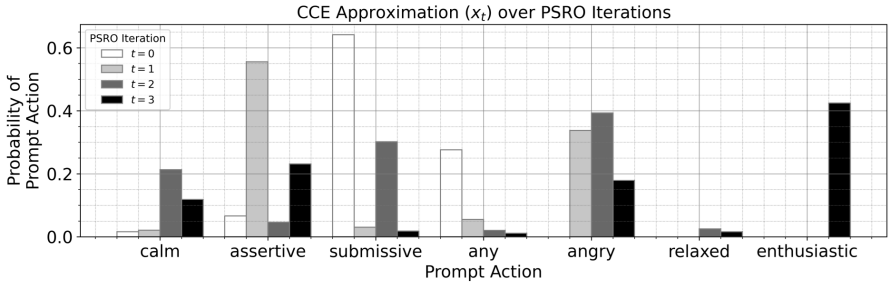


Figure 5 | Prompt-Space Response-Oracles (Algorithm 5) demonstration on fruit trading domain.

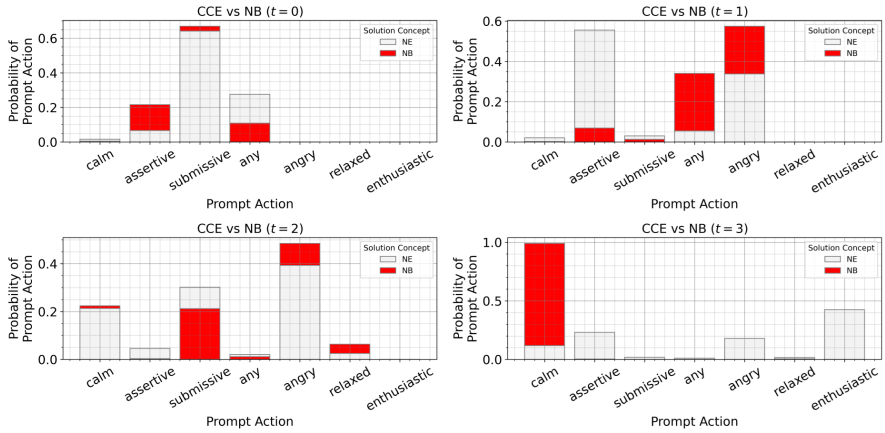


Figure 6 | The same PSRO run as Figure 5 but reporting the Nash bargaining solution in red at each iteration for the fruit trading domain.

For the fruit trading game "submissive" initially holds the most mass under the CCE ($t = 0$), however, it then gives way to more passionate tones such as "assertive", "angry", and "enthusiastic" that may benefit a more aggressive haggler. Inspecting Figure 6, it is interesting that "calm" is the final NB solution

whereas “assertive”, “angry”, and “enthusiastic” (and not “calm”) are the predominant actions under the CCE. Both players may extract higher collective value if they remain “calm” during negotiations.

Generalization Performance

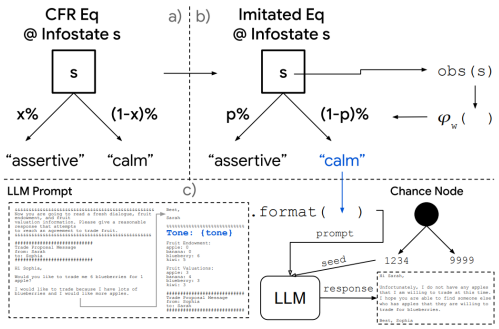


Figure 7 | Imitating Strategic Play. a) A single dialogue game is randomly generated and CFR is run to approximate a coarse-correlated equilibrium which constitutes a distribution over actions at every infostate in the game. b) An embedding of every infostate in the game, `obs(s)`, is stored along with the CFR equilibrium. The set of all paired embeddings and distributions constitutes the imitation learning dataset. A neural network $\psi_w : \texttt{obs}(s) \to \Delta$ is trained to imitate this dataset. c) ψ_w can then be used on newly generated games to produce distributions over actions. These actions, e.g., “calm”, are then used to format LLM prompts. Stochasticity of the LLM output is explicitly modeled with chance nodes that uniformly sample random seeds to pass to the LLM. Given the prompt and seed, the LLM generates a player response in natural language.

We generate 200 games using our procedural game generation approach above. For each game, we use 10 iterations of OpenSpiel’s built-in counterfactual regret minimization (CFR) to solve for an equilibrium. For each game, we save vector observations of each information state along with the optimal equilibrium policy returned by CFR for that infostate (imitation dataset). We then train a neural network policy to predict the equilibrium probabilities conditioned on the observations. Finally, we deploy this trained model against an LLM that only plays the action “any” on held out games.

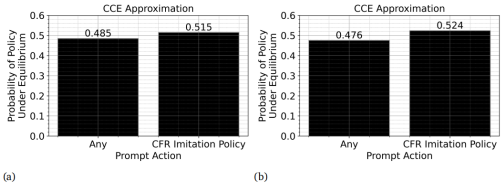


Figure 8 | Proof-of-Improvement: Equilibrium Evaluation of Imitation Learned Policy against Baseline LLM in (8a) scheduling a meeting and (8b) trading fruit.

We find that more mass lies on our CFR imitation policy under the equilibrium distribution implying it achieves higher payoff than the vanilla policy. Importantly, this implies our proposed approach results in an improved policy.

1.3 THE CONSENSUS GAME: LANGUAGE MODEL GENERATION VIA EQUILIBRIUM SEARCH

1.3.1 Introduction

Generative LM: A modelling sitribution $P_{LM}(y|x, correct)$
Discriminative LM: A modelling distribution $P_{LM}(v|y, x)$ where $v \in \{correct, incorrect\}$

1.3.2 The Consensus Game

A generator and a discriminator. Generator generates $v \in \{correct, incorrect\}$ and discriminator generates y, and produces a natural language string from a fixed set of candidates. Finally, this string is observed by the discriminator, who tries to guess the value of the correctness parameter by selecting

one of $\{correct, incorrect\}$ as an answer. Both players obtain a payoff of 1 if the discriminator correctly identifies the value of the correctness parameter, 0 otherwise.

$$u_G(\pi_G, \pi_D) = \frac{1}{2} \sum_{v \in \{corrcet, incorrect\}} \sum_{y \in \mathcal{Y}} \pi_G(y|v) \cdot \pi_D(v|y)$$

$$u_D(\pi_G, \pi_D) = \frac{1}{2} \sum_{v \in \{corrcet, incorrect\}} \sum_{y \in \mathcal{Y}} \pi_G(y|v) \cdot \pi_D(v|y)$$

We wish to find an optimal pair of policies (Nash equilibria). However, Nash equilbria of the CONSENSUS GAME are not guaranteed to provide the second criterion of reasonableness. This is because the CONSENSUS GAME admits a multitude of Nash equilibria that are incompatible with the commonsense notion of truthfulness.
Hence we add a regularization term to the utility functions, which penalize deviation from some pair of initial policies π_G^1 and π_D^1. The initial policies may be derived from an LM prompted with some initial string x, in order to obtain context-predictions.

$$u_G(\pi_G, \pi_D) = -\lambda_G KL\left[\pi_G(\cdot|v); \pi_G^1(\cdot|x, v)\right] + \frac{1}{2} \sum_{v \in \{corrcet, incorrect\}} \sum_{y \in \mathcal{Y}} \pi_G(y|v) \cdot \pi_D(v|y)$$

$$u_D(\pi_G, \pi_D) = -\lambda_D KL\left[\pi_D(\cdot|v); \pi_D^1(\cdot|x, v)\right] + \frac{1}{2} \sum_{v \in \{corrcet, incorrect\}} \sum_{y \in \mathcal{Y}} \pi_G(y|v) \cdot \pi_D(v|y)$$

Equilibrium Ranking: Generating text by performing no-regret learning in the CONSENSUS GAME (training-free consensus-planning method). Regret is the gap between the chosen action and the best action in hindsight.

Initial Policies:

$$\pi_G^1(y|x, v) \propto \frac{P_{LM}(y|x, v)}{\sum_{v'} P_{LM}(y|x, v')}$$
$$\pi_D^1(v|x, y) \propto \frac{P_{LM}(v|x, y)}{\sum_{y'} P_{LM}(v|x, y')}$$

Evolution of Policies: We find the Nash Equilibrium by local regret minimization. We find that the local regret minimization problems are composed of a bilinear term, plus a strongly convex KL-regularization term. Such composite utilities can be handled by the piKL algorithm, which is specifically designed to perform regret minimization on KL-regularized objectives.

$$Q_G^t(y|x, v) := \frac{1}{2t} \sum_{\tau=1}^{t} \pi_D^\tau(v|x, y)$$

$$Q_D^t(v|x, y) := \frac{1}{2t} \sum_{\tau=1}^{t} \pi_G^\tau(y|x, v)$$

Update:

$$\pi_G^{t+1}(y|x, v) \propto exp\left\{\frac{Q_G^t(y|x, y) + \lambda_G \log \pi_G^1(y|x, v)}{1/(\eta_G t) + \lambda_G}\right\}$$
$$\pi_D^{t+1}(v|x, y) \propto exp\left\{\frac{Q_D^t(v|x, y) + \lambda_D \log \pi_D^1(v|x, y)}{1/(\eta_D t) + \lambda_D}\right\}$$

piKL algorithm guarentees:

- Convergence to a Nash equilibrium
- Regularization towards reasonableness (average policy remais close to initial)
- Avoidance of regret

Equilibrium ranking returns π_G^* and π_D^*, which are the optimal policies for the generator and discriminator. We can then use these policies to generate text.

Experiments

Hyperparameters: $\eta_D = \eta_G = 0.1$, $\lambda_D = \lambda_G = 0.1$. Equilibrium ranking run for 5000 iterations.
For generative tasks, top 50 responses were taken. LLaMa 7B and 13B were used.
In the multiple-choice based datasets (ARC, RACE, HHH, MMLU), we consider the following approaches:

- Generative Ranking (G): It ranks every candidate y by $P_{LM}(y|x, correct)$ and picks the top candidate.
- Mutual Information Ranking (MI): It is an ensemble-based approach that reweights every candidate y by $P_{LM}(y \mid x, correct) \cdot P_{LM}(correct|x, y)$.
- Self-Contrastive Ranking (SC): This approach utilizes the normalized generator π_G^1 to reweight every candidate y by $\pi_G^1(correct|x, y)$.
- Discriminative Ranking (D): This approach reweights every query-candidate pair (x, y) by $\pi_D^*(correct|x, y)$.
- Equilibrium Ranking Generator (ER-G): Similar to SC, this approach utilizes the final EQUILIBRIUM-RANKING-based generator π_G^* to reweight every candidate y by $\pi_G^*(correct|x, y)$.
- Equilibrium Ranking Discriminator (ER-D): Similar to D, this approach utilizes the final EQUILIBRIUM-RANKING-based discriminator π_D^* to reweight every query-candidate pair (x, y) by $\pi_D^*(correct|x, y)$.

Results

Domain	Model	G*	G	MI	SC	D	Equil. ranking ER-G	Equil. ranking ER-D
MMLU	LLaMA-7B	–	30.4	33.1	30.5	**40.4**	39.4	39.9
	LLaMA-13B	–	41.7	41.8	41.7	41.9	44.9	**45.1**
ARC Easy	LLaMA-7B	72.8	68.2	68.8	69.5	52.5	**71.6**	71.5
	LLaMA-13B	74.8	71.2	71.5	73.0	65.0	76.1	**76.4**
ARC Challenge	LLaMA-7B	47.6	47.3	47.4	56.5	42.7	**58.7**	58.3
	LLaMA-13B	52.7	51.9	52.1	59.3	48.5	61.1	**61.4**
RACE Middle	LLaMA-7B	61.1	57.7	57.7	60.4	51.5	63.2	**63.5**
	LLaMA-13B	61.6	60.1	60.2	64.8	58.3	67.9	**68.6**
RACE High	LLaMA-7B	46.9	46.4	46.3	53.1	46.0	56.3	**56.4**
	LLaMA-13B	47.2	47.9	48.4	58.9	55.1	62.1	**62.8**
HHH	LLaMA-7B	–	59.3	57.9	67.4	70.1	**71.5**	**71.5**
	LLaMA-13B	–	60.2	59.7	57.9	**69.2**	61.1	61.1

- MMLU: For both LLaMA7B and LLaMA-13B, the EQUILIBRIUM-RANKING-based approaches matches or outperforms all other baselines. In fact, zero-shot LLaMA-7B with ER-D (39.9) outperforms 5-shot LLaMA-7B (35.1), while zero-shot LLaMA-13B with ER-D (45.1) is competitive with 5-shot LLaMA-13B (46.9). LLaMA-7B with ER-D (39.9) even outperforms zero-shot GPT3-175B (37.7), while zero-shot LLaMA-13B with ER-D (45.1) outperforms 5-shot GPT3-175B (43.9).
- ARC: On ARC-Easy, ER-D outperforms our implementation of generative ranking. We also note that LLaMA-13B with ER-D (76.4) outperform all the baseline approaches and is even competitive with the much larger PaLM-540B model (76.6). On ARC-Challenge, ER-D significantly outperforms all the baseline approaches. We also note that LLaMA-7B with ER-D (58.3) and LLaMA-13B with ER-D (61.4) outperforms even the much larger models: LLaMA-65B (56.0) and PaLM-540B (53.0) by up to 8%.
- RACE: On RACE-middle, like before, ER-D based models outperforms all the baselines. We note that LLaMA-13B with ER-D (68.6) even outperforms much larger models: LLaMA-65B (67.9) and PaLM-540B (68.1). On RACE-high, we have a similar story as with ARC-C. ER-D outperforms all baselines. LLaMA-7B with ER-D (56.4) is able to significantly outperform much larger models: LLaMA-65B (51.6) and PaLM-540B (49.1).
- HHH: LLaMA-7B with ER-D outperforms baselines; although LLaMA-13B with ER-D with the default parameter performs worse than discriminative ranking (D) (69.2), ER-D with $\lambda_G = 0.01$ and $\lambda_D = 1.0$ achieves an accuracy of 70.6%.

Domain	Model	Greedy	MI	SC	D	**Equil. ranking** ER-G	ER-D
TruthfulQA	LLaMA-7B	33.41	34.79 ±0.90	**34.91** ±0.57	34.17 ±1.19	34.61 ±0.99	34.27 ±0.39
	LLaMA-13B	33.05	36.30 ±0.37	34.61 ±1.33	39.05 ±1.42	**39.83** ±2.20	38.63 ±1.76

TruthfulQA: On this task, we consider greedy decoding in addition to our other ranking-based approaches. In this setting, 10 candidate sequences are sampled using nucleus and top-k sampling. These candidates are then ranked based on the approaches we described earlier.For a sequence a, the BLEU-Acc over reference correct candidates $b_{correct}$ and reference incorrect candidates $b_{correct}$ is computed as follows:

$$BLEU - Acc(a) := \mathbb{I}(BLEU(a, b_{correct}) > BLEU(a, b_{incorrect}))$$

With LLaMA-7B, we observe only modest improvements for ER-G and ER-D over the greedy baseline. However, with LLaMA-13B, we note increased scores for both methods. This benchmark is known to exhibit negative scaling (performance drop as the model size increases). The performance difference with ER-G between LLaMA-7B and LLaMA-13B shows that EQUILIBRIUM-RANKING is in fact capable of mitigating this behavior.

Domain	Model	Greedy	MV	MI	SC	D	**Equil. ranking** ER-G	ER-D
GSM8K	LLaMA-7B	10.8	14.7 ±0.2	14.6 ±0.5	13.4 ±0.3	15.0 ±0.6	13.0 ±0.5	**15.1** ±0.6
	LLaMA-13B	14.9	22.5 ±0.5	22.5 ±0.8	**23.1** ±0.5	22.5 ±0.6	22.5 ±0.6	23.0 ±0.5

GMS8K: We generate 20 candidate reasoning paths sampled using nucleus and top-k using the 8-shot setup. We employ self-consistency over the candidate sequences, where we score each reasoning path with our baselines. Finally, we aggregate the scores for each answer corresponding to the reasoning paths and pick the answer with the highest score. We note that EQUILIBRIUM-RANKING-based approaches performs on par or slightly better compared to self-consistency (majority vote).

1.3.3 Results

The application of EQUILIBRIUM-RANKING-based approaches consistently yields improved results, surpassing or at least matching the performance of all baseline approaches across various tasks. This robustness is particularly interesting, as it demonstrates that EQUILIBRIUMRANKING is adept at handling diverse scenarios, even in situations when the initial GENERATOR or DISCRIMINATOR are not effective. Finally, we note that EQUILIBRIUM-RANKING demonstrates computational efficiency by eliminating the need for repetitive queries to language models.

2 Racial Bias

2.1 Racial Bias within Face Recognition: A Survey

2.1.1 Introduction

A face recognition system comprises a training set D_{train} and a test set D_{test}. Any face dataset can be formed as $D = \{X, Y, S\}$ where X denotes the set of images, Y denotes the set of subject labels, and S denotes the set of sensitive race or race-related labels. A mapping function f plays a significant role in face recognition systems as it maps any given image x into the feature embedding vector z. f is selected from a function space Ω via a loss function $\mathcal{L}$ which measures the performance of a given training set, D_{train}, for any of the aforementioned face recognition tasks.

$$f^* = \underset{f \in \Omega}{argmin}(\mathcal{L}_{softmax}(f))$$

Metrics:

- $$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$
- $$TMR = \frac{TP}{TP + FN}$$
- $$FMR = \frac{FP}{FP + TN}$$
- $$TNMR = \frac{TN}{TP + FN}$$
- $$FNMR = \frac{FN}{FP + TN}$$
- ROC curve, plots TMR against FMR at different thresholds.
- Cosine similarity between embedding vectors.

Four most used fairness definitions:

1. Fairness Through Unawareness requires that a machine learning algorithm have an independent conditional probability P of the output given X from S (racial labels).

$$P(Y|X) = P(Y|X, S)$$

2. Individual Fairness refers to treating similar individuals coequally, meaning that an algorithm is fair if it gives similar predictions to similar individuals.

3. Group fairness (or Statistical parity / Demographic parity) enforces the predicted subject labels $\hat{Y}$ to be independent of S which can be denoted $P(\hat{Y}|S = s) = P(\hat{Y}|S = s), s \in \{0, 1, \ldots, r\}$ where r is the number of different sensitive race labels in the set.

4. Equal Opportunity, (or Equalised Odds) is satisfied if an algorithm predictions $\hat{Y}$ is independent of S conditioned on Y.

Causes of raical bias in data:

- Race: Subject face images form the primary information source that encapsulates these race-related biological and physical differences, which are then combined with additional information, including gender, age, pose, facial expression and contextual aspects such as scene background, illumination, subject clothing and facial accessories such as glasses, facial hair, jewellery and makeup. On this basis, it becomes possible to adopt any such ideology via the use of racial groupings and classifications that are introduced to face recognition with the aim of quantifying racial bias. y four specific problems with the racial categories: (1) categories are not clearly defined and are often loosely associated with geographic origin, (2) categories that are extremely broad, with continent-spanning construction that results in individuals with vastly different physical appearance and ethnic backgrounds being grouped incongruously into the same racial category, (3) categories narrow down the differences between ethnic groups with distinct languages, cultures, separation in space and time, and phenotype into the same racial category. (4) assigning a single racial category to a face example for performance evaluation of any form of automated analysis, including face recognition, is not an ideal solution as it cannot capture a substantial proportion of the distribution of diversity and variation within the human race.

- Skin Tone: Skin tone scale grouping strategies alone carry various concerns for the mitigation of racial bias within face recognition:

  - Erroneous Skin Tone Annotation: Most skin tone scales are designed to measure skin tone on physical human subjects in a medical or dermatological context. By contrast, face recognition systems instead used such annotations for digitally captured face images that form part of the training and test data sets. Scene illumination, camera characteristics, demographic characteristics (race, age, gender), and other factors (make-up, wearing glass, hairstyle, head pose), make it difficult to accurately measure skin tone in face images.
  - Narrow Representation of Scales: The most commonly used skin tone scales used for accessing aspects of racial bias are either too narrow in terms of their discretisation of the skin tone spectrum to facilitate capture of the foundational reasons for bias or alternatively offer the less representative capability for specific groups.
  - Skin Tone as a Single Dimension of Race: Solely aligning racial grouping with skin tone only transforms the racial bias problem into a single-faceted problem. Moreover, there is no clear evidence that skin tone alone is the primary driver for disparate false match rates within

face recognition performance. considering other race-related facial attributes, including lips, eye, and face shape when measuring racial bias in this context in order to enable improved interpretation and derivation of bias factors.

Binary Skin-Tone Scale: . In order to model skin colour on imagery, several studies proposed quantitative colour-space divisors (i.e. a dark-light pixel colour threshold) and simply grouped skin colours into binary categories. In the racial bias context, many studies adopt such a darker-lighter skin tone grouping by either narrowing the Fitzpatrick scale or dividing subject skin tone variance into binary categories.
Fitzpatrick Scale: The dermatologist Thomas B. Fitzpatrick developed his Fitzpatrick Skin Tone Scale to assess the propensity of the skin to burn during photo-therapy (i.e. the treatment of skin conditions using intense ultra-violet light sources). Initially, four different types ranging from Type I (always burns, does not tan) to Type IV (rarely burns, tans with ease) were released. Later, he extended his scale to include a broader range of skin types (Type V and IV) in order to offer a more granular representation across darker skin tones.
Individual Typology Angle (ITA): This method utilises the reflection of skin light via spectrophotometers that measure LaB colour values of the skin (L: Lightness. a: Red/Green Value. b: Blue/Yellow Value) to represent the intensity of pigments such as carotene, haemoglobins, phaeomelanin, and eumelanin. Accordingly, Chardon proposes six physiologically skin categories: very light, light, intermediate, tan, brown, and dark estimated via equation of ITA $ITA = \arctan\left(\frac{L-50}{b}\right) \times \frac{180}{\pi}$.
Monk Scale: Most recently, the work of Ellis Monk produced a new 10-shade skin tone scale designed to facilitate the construction of more representative datasets for the development of on-line consumer services.

- Facial Phenotype: Racial appearance bias as a negative disposition toward phenotypic variations in facial appearance. Individuals with more stereotypical racial appearance suffer from poorer socioeconomic outcomes than those with less stereotypical appearance for their race. A set of race-related facial phenotype attributes such as skin tone, nose shape, and lip shape are of primary interest for quantifying and addressing racial bias in face recognition. Compared to the prevalence of race or skin tone categories, phenotype-based groupings have received less attention across the racial bias literature to date, as they involve both skilled attribute labelling for dataset construction and a significantly more complex evaluation strategy due to the significant number of phenotype categories, and phenotype combinations present.

2.1.2 Sources of Bias:

Image Acquisition:

- Facial Imaging: Concerns about privacy, fairness, freedom and autonomy, and security.
- Dataset Curation: Such a sampling process is often affected by sampling bias, which significantly impacts racial bias in face recognition.
- Dataset Bias Mitigation:
  - Selection Bias which is same as sampling bias
  - Capture Bias occurs when the dataset imagery contains targets (faces) that have minimal spatial and illumination variation and can be related with pose bias within face recognition context, as there is still poor pose variance within datasets

- Category/Label Bias poses the ill-definition or mislabelling of subject identities and racial categories
- Negative Set Bias defines bias against target appearances that are not represented in the data set leading to recognition models that are overconfident and misrepresent performance by considering only a skewed subset of possible real-world data samples.

Face Localisation

Corrupted data is more likely to cause face detection errors in specific demographic groups. Confounding factors including facial orientation, illumination, and resolution, can cause such disparate performance among demographic groups. The presence of imaging, sampling and dataset bias within these face detection benchmark datasets again translates through the subsequent stages of face recognition resulting in skewed overall face recognition pipeline performance.

Facial Representation

- Backbone Architectures: Use of Diffusion-Convolutional Neural Networks (DCNNs). Each layer t consists of K kernels with weights $W = W_1, W_2, \ldots, W_K$ and added bias filters $B = b_1, \ldots, b_K$. Subsequently, each layer applies an element-wise nonlinear transform to generate multiple feature map representations and passes the result to the next layer $x_t = \sigma(W_K \cdot x_{t-1} + b_K)$. Moreover, at the end of each layer, a pooling function down-samples the feature maps by taking the maximum or average value of adjacent pixels (patch).
- Baseline Loss Functions: Good and easy to compute loss functions help in face recognition. cosface used:

$$\mathcal{L}_{cosface} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{||z||(W_j^T z_i - m)}}{e^{||z||(W_j^T z_i - m)} + \sum_{j\neq y_i}^{n} e^{||z||(W_j^T z_i)}}$$

  arcface used:

$$\mathcal{L}_{arcface} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{||z||(W_j^T z_i + m)}}{e^{||z||(W_j^T z_i + m)} + \sum_{j\neq y_i}^{n} e^{||z||(W_j^T z_i)}}$$

  Strongly dependent on choice of hyperparameters.
- Mutual Information Mitigation: The high mutual information between facial identity and underlying racial features within face images generally transfer into the learned feature embedding of contemporary DCNN based techniques and hence results in an unsatisfied fairness through unawareness criteria. Several techniques have been proposed to mitigate this issue, including adversarial learning, fairness-aware loss functions, and fairness-aware training strategies. Recent methods use knowledge distillation.
- Loss Function Based Mitigation: Focuses on setting adaptive margins to tackle racial bias. To fix this, the authors propose a novel loss function where the margin m is adaptively set based on the racial label s of the input image. Another approach, Asymmetric Rejection Loss aims to reduce the racial bias within trained face recognition models by taking advantage of unlabelled images of under-represented groups.
- Domain Adaptation Based Mitigation: These techniques use multiple labelled source domains with different distributions to improve generalisation to new target datasets. Meta Learning: Synthesise the source/target domain and force the model to learn effective representations of both synthesised source and target domains. Cross-Domain Triplet (CDT) loss uses similarity metrics from one domain to learn compact feature clusters of identities by incorporating them into another domain.

Face Verification
Face verification performance is measured in terms of accuracy and matching rates over pairs of identical/nonidentical subject images in order to evaluate the number of correct identities matches over all the set of all paired images presented. In order to confirm a match, the feature embedding vector z_{target} from a presented unseen subject. image instance x_target, and those of a subject image $x_{reference}$ held on record a priori, $z_{reference}$, are compared using a distance or similarity score across the learnt feature embedding space. An apriori *threshold* is used to make a decision on the similarity of $z_{target} \approx z_{reference}$ such that a verified identity can be confirmed or not. Significant performance on face verification on public benchmark datasets where the racial diversity within these datasets is often limited, biased and overlooked

Face Identification
The process involves comparing an obtained query face image x_{target} with a large database of reference images $X_{enrolment}$. Face identification tasks can be sub-categorised as either closed-set, when the target is always in the enrolment set ($x_{target} \in X_{enrolment}$), or open-set, when the target may or may not be in the enrolment set ($x_{target} \in X_{enrolment}$ or $x_{target} \notin X_{enrolment}$). In order to perform a closed-set face identification task, a multi-class classifier is used to identify the target image x_{target} via the use of feature embedding vector z_{target} over $Z_{enrolment}$.

Evaluation Biases:

- Evaluation bias encourage the development of models that only perform well on the specific racial groupings as the per distribution of the dataset. Evaluation bias is also related to the decisions made at this stage of the face recognition pipeline, including pairing selection, threshold optimisation, distance and normalisation functions. performance is highly dependent on the number of images available in a template. Performance is highly dependent on the number of images available in a template.

- Adversarial Gender De-biasing algorithm (AGENDA) trains a shallow network that removes the gender information of the embeddings extracted from a pre-trained network, with PASS to deal with any sensitive attribute and proposed a novel discriminator training strategy.

- Fair Template Comparison (FTC) method replaces the computation of the cosine similarity score by an additional shallow neural network trained using cross-entropy loss, with a fairness penalisation and L2 penalty term to prevent over-fitting.

- Group-specific threshold (GST) in which the sensitive attributes themselves define its calibration sets.

- Fair Score Normalisation (FSN) method, which is essentially GST with unsupervised clusters. FSN normalises the scores by requiring the model FMRs across unsupervised clusters to be the same predefined global FMR.

- Fairness Calibration (FairCal) method that applies the K-means algorithm to the image feature representation vectors Z and makes partitions of the embedding space into K clusters. For each set, it calculates separate calibration map scores to cluster-conditional probabilities of the set. If the pair of images belong to the same subject cluster, the algorithm uses the score; if not, it uses the weighted average of the calibrated scores in each cluster of corresponding image features.

Open-set face identification requires a threshold to report a match or non-matched decision over test target imagery. Two types of errors in face identification false-non-matched identification and false-matched

identification together with their dependency on a threshold that defines the minimum similarity required to report a match.
Designing an ideal evaluation strategy is yet another crucial step in the face recognition processing pipeline. This step becomes particularly important in order to address racial bias within face recognition, as every decision made at this stage can have a significant impact on the overall performance and performance across different groups.

3 Diplomacy

3.1 Introduction

In the game, seven players compete for majority control of 34 "supply centers" (SCs) on a map. On each turn, players simultaneously choose actions consisting of an order for each of their units to hold, move, support or convoy another unit. If no player controls a majority of SCs and all remaining players agree to a draw or a turn limit is reached then the game ends in a draw. In this case, we use a common scoring system in which the score of player i is $C_i^2 / \sum_{i'} C_{i'}^2$, where C_i is the number of SCs player i owns.
Most recent successes in no-press Diplomacy use deep learning to imitate human behavior given a corpus of human games. Self-play approaches, actor-critic approaches, one-ply search, etc. have been used so far.
Regularizing inference-time search techniques can produce agents that are not only strong but can also model human behaviour well. In the domain of no-press Diplomacy, they show that regularizing hedge with a KL-divergence penalty towards a human imitation learning policy can match or exceed the human action prediction accuracy of imitation learning while being substantially stronger. **Markov Game:** Interactive game where, actions are dependent only on the current state and not on the history of the game.
HEDGE: It's an iterative algorithm that converges to a Nash equilibrium.

$$Q^t(a_i) = \frac{1}{t} \sum_{t' \leq t} u_i(a_i, a_{-i}^{t'})$$

$$\pi_i^t(a_i) \propto exp(Q^{t-1}(a_i)/\kappa_{t-1})$$

Regret Matching algorithm is not used here.
It's inspired from DORA algorithm (an algorithm that is similar to past model-based reinforcement-learning methods) except in place of Monte Carlo tree search, which is unsound in simultaneous-action games such as Diplomacy or other imperfect information games, it instead uses an equilibrium-finding algorithm such as hedge or RM to iteratively approximate a Nash equilibrium for the current state.
A deep neural net trained to predict the policy is used to sample plausible actions for all players to reduce the large action space in Diplomacy down to a tractable subset for the equilibrium-finding procedure, and a deep neural net trained to predict state values is used to evaluate the results of joint actions sampled by this procedure. The idea of Nash-Q was used to compute equilibrium policies σ in a subgame where the actions correspond to the possible actions in a current state and the payoffs are defined using the current approximation of the value function.
A version of piKL algorithm (piKL-hedge) was used for behavioral cloning. The hyperparameter λ was found to be better samples from a distribution β over λs, called DIL-piKL (Distribution Lambda piKL).

Algorithm 1: DiL-piKL (for Player i)

Data:
- A_i set of actions for Player i;
- u_i reward function for Player i;
- Λ_i a set of λ values to consider for Player i;
- β_i a belief distribution over λ values for Player i.

function INITIALIZE()
- $t \leftarrow 0$
- **for each** action $a_i \in A_i$ **do**
  - $Q_i^0(a_i) \leftarrow 0$

function PLAY()
- $t \leftarrow t + 1$
- sample $\lambda \sim \beta_i$
- let $\boldsymbol{\pi}_{i,\lambda}^t$ be the policy such that

$$\boldsymbol{\pi}_{i,\lambda}^t(a_i) \propto \exp\left\{\frac{Q^{t-1}(a_i) + \lambda \log \tau_i(a_i)}{\kappa_{t-1} + \lambda}\right\}$$

- sample an action $a_i^t \sim \boldsymbol{\pi}_{i,\lambda}^t$
- play $a_i^t \in A_i$ and observe actions $\boldsymbol{a}_{-i}^t$ played by the opponents
- **for each** $a_i \in A_i$ **do**
  - $Q^t(a_i) \leftarrow \frac{t-1}{t} Q^{t-1}(a_i) + \frac{1}{t} u_i(a_i, \boldsymbol{a}_{-i}^t)$

Theoretical Results:

Theorem 1: Let W be a bound on the maximum absolute value of any payoff in the game, and $Q_i := \frac{1}{n} \sum_{a \in A_i} \log \tau_i(a)$. Then, for any player i, type $\lambda_i \in \Lambda_i$, and number of iterations T, the regret cumulated can be upper bounded as

$$\max_{\pi \in \Delta(A_i)} \left\{ \sum_{t=1}^{T} \tilde{u}_{i,\lambda_i}(\pi, a_{-1}^t) \right\} \leq \frac{W^2}{4} min\{\frac{2 \log T}{\lambda_i}, T\eta\} + \frac{\log \eta_i}{\eta} + \rho_{i,\lambda_i}$$

where $\rho_{i,\lambda_i} := \lambda_i(\log \eta_i + Q_i)$.

Theorem 2: When both players in a 2 player 0 sum game learn using DiL-piKL for T iterations, their policies converge almost surely to the unique Bayes-Nash equilibrium (π_{i,λ_i}^*) of the regularized game defined by utilities $\tilde{u}_{i,\lambda_i}$.

3.2 Model

By replacing the equilibrium-finding algorithm used in DORA with DiL-piKL, we obtain a new algorithm named RL-DiL-piKL, which can learn a strong and human-compatible policy as well as a value function that can accurately evaluate game states, assuming strong and human-like continuation policies. It's trained with the same loss function as DORA with a λ distribution. At evaluation time we perform 1-ply lookahead where on each turn we sample up to 30 of the most likely actions for each player from the RL policy proposal network. The model will be called Diplodocus.

3.3 Experiments

Algorithms:

- Diplodocus-Low: $\beta_i \sim Unif\{10^{-4}, 10^{-1}\}\forall i$
- Diplodocus-High: $\beta_i \sim Unif\{10^{-2}, 10^{-1}\}\forall i$
- DORA is an agent that is trained via self-play and uses RM as the search algorithm during training and test-time.
- DNVI is similar to DORA, but the policy proposal and value networks are initialized from human BC pretraining.
- DNVI-NPU is similar to DNVI, but during training only the RL value network is updated. The policy proposal network is still trained but never fed back to self-play workers, to limit self-play drift from human conventions.
- BRBot is an approximate best response to the BC policy. It was trained the same as Diplodocus, except that during training the agent plays one distinguished player each game with $\lambda = 0$ while all other players use $\lambda = \infty$.
- SearchBot, a one-step lookahead equilibrium search agent.
- HedgeBot is an agent similar to SearchBot but using our latest architecture and using hedge rather than RM as the equilibrium-finding algorithm.
- FPPI-2 and SL are two agents.

Algo Tournament: These games used a time limit of 5 minutes per turn and a stochastic game-end rule where at the beginning of each game year between 1909 and 1912 the game ends immediately with 20% chance per year, increasing in 1913 to a 40% chance.

Agent	**Score against population**
Diplodocus-Low	29% ± 1%
Diplodocus-High	28% ± 1%
DNVI-NPU (retrained) (Bakhtin et al., 2021)	20% ± 1%
BRBot	18% ± 1%
DNVI (retrained) (Bakhtin et al., 2021)	15% ± 1%
HedgeBot (retrained) (Jacob et al., 2022)	14% ± 1%
DORA (retrained) (Bakhtin et al., 2021)	13% ± 1%
FPPI-2 (Anthony et al., 2020)	9% ± 1%
SearchBot (Gray et al., 2020)	7% ± 1%
SL (Anthony et al., 2020)	6% ± 1%

Table 1: Performance of different agents in a population of various agents. Agents above the line were trained using identical neural network architectures. Agents below the line were evaluated using the models and the parameters provided by the authors. The ± shows one standard error.

Human Tournament: Each game had exactly one agent and six humans. The players were informed that there was an AI agent in each game, but did not know which player was the bot in each particular game. 5-minute long games were played.

	Rank	Elo	Avg Score	# Games
Diplodocus-High	1	181	27% ± 4%	50
Human	2	162	25% ± 6%	13
Diplodocus-Low	3	152	26% ± 4%	50
Human	4	138	22% ± 9%	7
Human	5	136	22% ± 3%	57
BRBot	6	119	23% ± 4%	50
Human	7	102	18% ± 8%	8
Human	8	96	17% ± 3%	51
...	...	...	...	...
DORA	32	-20	13% ± 3%	50
...	...	...	...	...
Human	43	-187	1% ± 1%	7

Table 2: Performance of four different agents in a population of human players, ranked by Elo, among all 43 participants who played at least 5 games. The ± shows one standard error.

In addition to the tournament, we asked three expert human players to evaluate the strength of the agents in the tournament games based on the quality of their actions. Games were presented to these experts with anonymized labels so that the experts were not aware of which agent was which in each game when judging that agent's strategy.

Results: For algo tournament, Diplodocus-Low and Diplodocus-High perform the best by a wide margin.

For human tournament, results show that Diplodocus-High performed best among all the humans by both Elo and average score. Diplodocus-Low followed closely behind, ranking second according to average score and third by Elo. BRBot performed relatively well, but ended ranked below that of both DiL-piKL agents and several humans. DORA performed relatively poorly. Two participants achieved a higher average score than the Diplodocus agents, a player averaging 35% but who only played two games, and a player with a score of 29% who played only one game.

For expert evaluation, All the experts picked a Diplodocus agent as the strongest agent, though they disagreed about whether Diplodocus-High or Diplodocus-Low was best. Additionally, all experts indicated one of the Diplodocus agents as the one they would most like to cooperate with in a game.

3.4 RL Training

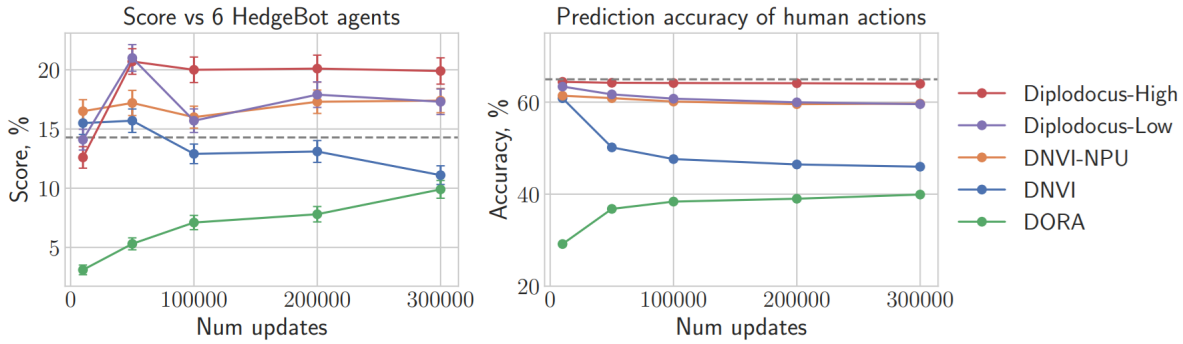


Figure 3: Performance of different agents as a function of the number of RL training steps. **Left:** Scores against 6 human-like HedgeBot agents. The gray dotted line at score $1/7 \approx 14.3\%$ corresponds to tying HedgeBot. The error bars show one standard error. **Right:** Order prediction accuracy of each agent's raw RL policy on a held-out set of human games. The gray dotted line corresponds to the behavioral cloning policy. **Overall:** Diplodocus-High achieves a high score while also maintaining high prediction accuracy. Unregularized agents DNVI and DORA do far worse on both metrics.

We compare different RL agents across the course of training by varying the training methods for the value and policy proposal networks, but using the same search setting at evaluation time.
We found that agents without biasing techniques (DORA and DNVI) diverge from human play as training progress. By contrast, Diplodocus-High achieves significant improvement in score while keeping the human prediction accuracy high.

3.5 Compressed Models Decompress Race Biases: What Quantized Models Forget for Fair Face Recognition

Trying to use synthetic data to fix racial biases.

Datasets

- **MS1MV2:** MS1MV2 is widely used in the literature to train and compare several deep face recognition models. It is a refined version of the original MS-Celeb-1M dataset, which further improved the training of these systems. The dataset contains 85k different identities and almost six million images and it is not balanced with respect to the race.
- **BUPT-Balancedface and BUPT-Globalface:** These datasets have been created to mitigate race bias on face recognition through skin tone labelling as African, Asian, Caucasian and Indian. BUPTGlobalface contains two million images from 38k different identities, and the distribution of races follows their distribution in the world. On the other hand, BUPT-Balancedface contains 1.3 million images from 28k identities which are divided into 7k identities per race.
- **Synthetic Data:** Approximately 500k unlabelled synthetic images. These images have been generated by a generative adversarial network. Noise used as input to generate the images was sampled from a Gaussian distribution and fed to a pretrained generator.
- **LFW:** Racial Faces in-the-wild (RFW) was designed as a benchmarking dataset for fair face verification. Similarly, it includes labels for ethnicity, which allows for a fair assessment of potential biases. It contains 3000 individuals with 6000 image pairs for face verification.

3.6 Methods

First we checked if there's any bias in the various versions of QuantFace using MS1MV2 dataset.
Next a ResNet-34 was trained on BUPT-Balancedface and a second ResNet-34 was trained on BUPT-Globalface. We further trained a ethnicity classifier on BUPT-Balancedface to estimate the ethnicity distribution in the synthetic data. This classifier comprises a fully-connect layer on top of a pretrained Elastic-Arc model [1] model and achieves accuracies above 95%.

3.6.1 Experiments

The models were trained and the performance of the evaluated models was measures in terms of accuracy. For the fairness evaluation of these models we have utilised two metrics: the standard deviation between the different accuracies (STD), and the skewed error ratio (SER).

3.6.2 Results

TABLE COMPRISING THE RESULTS, EVALUATED ON RFW, FROM THE DIFFERENT MODELS TRAINED ON MS1MV2 AND THEIR RESPECTIVE QUANTIZED VERSIONS FOR DIFFERENT BITS AND QUANTIZATION STRATEGIES (REAL OR SYNTHETIC DATA). THE VERSIONS OF THE MODELS QUANTIZED WITH SYNTHETIC DATA SEEM TO DISPLAY BETTER FAIRNESS METRICS AT A COMPARABLE AVERAGE PERFORMANCE.

Model	Bits	Quant.	Caucasian	Indian	Asian	African	Avg.	STD	SER
MobileFaceNets	32	-	95.18%	92.00%	89.93%	90.22%	**91.83%**	2.41	2.09
	8	Real	95.32%	91.60%	89.27%	90.08%	91.57%	2.68	2.29
	8	Synth.	94.18%	91.83%	88.85%	89.72%	91.15%	**2.38**	1.92
	6	Real	90.05%	86.52%	82.88%	83.18%	85.66%	3.36	1.72
	6	Synth.	89.97%	86.95%	83.13%	84.40%	86.11%	3.02	**1.68**
ResNet-18	32	-	97.48%	95.38%	93.72%	94.27%	**95.21%**	1.66	2.49
	8	Real	97.42%	95.33%	93.55%	94.20%	95.13%	1.70	2.50
	8	Synth.	96.95%	95.07%	93.30%	93.87%	94.80%	**1.61**	**2.20**
	6	Real	96.93%	94.65%	92.52%	93.22%	94.33%	1.95	2.44
	6	Synth.	96.80%	94.78%	92.35%	93.28%	94.30%	1.94	2.39
ResNet-50	32	-	99.00%	98.15%	97.62%	98.32%	98.27%	**0.57**	**2.38**
	8	Real	99.07%	98.07%	97.65%	98.40%	**98.30%**	0.60	2.53
	8	Synth.	99.02%	97.72%	97.33%	97.88%	97.99%	0.73	2.72
	6	Real	98.32%	96.27%	94.55%	95.87%	96.25%	1.56	3.24
	6	Synth.	97.95%	96.63%	94.97%	96.20%	96.44%	1.23	2.45
ResNet-100	32	-	99.65%	98.88%	98.50%	99.00%	**99.01%**	**0.48**	4.29
	8	Real	99.57%	98.87%	98.15%	98.77%	98.84%	0.58	4.30
	8	Synth.	99.37%	98.72%	98.13%	98.78%	98.75%	0.51	2.97
	6	Real	95.27%	93.15%	90.32%	91.70%	92.61%	2.12	2.05
	6	Synth.	95.93%	93.40%	91.92%	92.60%	93.46%	1.75	**1.99**

Smaller models tend to have higher biases and lower performance in terms of average accuracy. ResNet-100 is an exception and this difference might be connected to the fact that SER becomes highly sensitive when the errors are below 1%. The quantized version of these models seems to retain the performance and bias advantaged when compared to simpler models. However, the usage of synthetic data has shown, for all the different precisions, a capability to reduce the bias while retaining the performance. From this data, it is not clear if the improvement is due to a specific characteristic of the synthetic data.

TABLE COMPRISING THE RESULTS, EVALUATED ON RFW, FROM TWO RESNET-34 MODELS TRAINED ON BUPT-GLOBALFACE (GL) AND THEIR RESPECTIVE QUANTIZED VERSIONS FOR DIFFERENT BITS AND QUA SYNTHETIC DATA). THE VERSIONS OF THE MODELS QUANTIZED WITH SYNTHETIC DATA SEEM TO PE

Train Data	Bits	Quant.	Caucasian	Indian	Asian	African	Av
BL	32	-	96.60%	94.50%	94.03%	93.37%	**94**
	8	BL	94.98%	93.60%	92.77%	90.95%	93
	8	Synth.	96.03%	94.40%	93.97%	92.50%	94
	6	BL	89.22%	87.87%	86.25%	82.80%	86
	6	Synth.	94.58%	92.88%	91.45%	91.13%	92
GL	32	-	97.67%	95.52%	94.15%	93.87%	**95**
	8	BL	95.42%	92.75%	91.83%	89.88%	92
	8	GL.	94.70%	92.15%	90.23%	88.75%	91
	8	Synth.	97.33%	95.15%	94.17%	93.55%	95

Training two ResNet-34 on BUPT-Balancedface and BUPT-Globalface shows, at full precision, that despite a higher performance of the latter, the balance of the former is essential to ensure better bias metrics. On the model trained with the BUPT-Balancedface the versions quantized with synthetic data has not only kept the same tendency, but it has also surpassed by a large margin the version of the method quantized with the balanced data. The ResNet-34 trained on the BUPT-Globalface performs better if quantized with the data from the BUPT-Balancedface instead of using the data from training.

4 piKL

4.1 Dec-POMDP

A Dec-POMDP is N agents indexed by $(1, \ldots, N)$, a state space $\mathcal{S}$, a joint action space $\mathcal{A} = \mathcal{A}^1 \times \ldots \times \mathcal{A}^N$, a transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$, a reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ and a set of observation function $o^i = \Omega^i(s), s \in S \forall i$. Define a trajectory of true states until time step t as $\tau_t = (s_0, a_0, r_0, \ldots, s_t)$, and it's AOH (action-observation history) for agent i as $\tau_t^i = (o_0^i, a_0, r_0, \ldots, o_t^i)$. Agent policy $\pi^i(\tau_t^i) = P(a_t^i|\tau_t^i)$ maps each possible AOH to a distribution over the action space of that agent.

Here a recurrent network is trained to model the expected total return for each action given the input AOH, $Q(\tau_t^i, a) = \mathbb{E}_{\tau \sim P(\tau_t|\tau_t^i)} R(\tau_t)$ where $R(\tau_t) = P_t' \geq t\gamma(t' - t)r_t$ is the sum of discounted future reward by unrolling the joint policy π on the sampled true game trajectory until termination. The joint policy is the greedy policy derived from each agent's Q-function.

SPARTA for search. Search agent i keeps track of the belief $\mathcal{B}(\tau_t^i) = P(\tau_t|\tau_t^i, \pi^{-1})$, which is the distribution of the trajectory of states given the AOH and partners' policies.

$$a_t^i = \underset{a}{argmax} Q_\pi(\tau_t^i, a) = \underset{a}{argmax} \mathbb{E}_{\tau_t \sim \mathcal{B}(\tau_t^i)}[r(\tau_t, a) + R_\pi(\tau_{t+1})]$$

This learned belief search technique is used in piKL. The acctions are sampled from:

$$P(a) \propto \pi_{anchor\ policy}(a|\tau_t^i) \cdot exp\left[\frac{Q_{\pi_{roll(\text{Output of a search algorithm})}}(\tau_t^i, a)}{\lambda}\right]$$

4.2 piKL algorithm

Algorithm 1 piKL-IL: modeling human with different skill levels. $P(\lambda)$ can be a discrete uniform distribution over a set of values or over a set of Gaussian distributions centered around those values. piKL-LBS(λ_i, π_{roll}, $\hat{\mathcal{B}}$, π_{anc}) is a function to act following Eq. 2 or its greedy variant. It samples from the learned approximate belief model $\tau_t \sim \hat{\mathcal{B}}(\tau_t^i)$ to estimate $Q_{\pi_{\text{roll}}}$.

procedure PIKL-IL(π_{BC}, $P(\lambda)$, k, d)
  ▷ π_{BC}: behavioral cloning policy trained from human data;
  ▷ $P(\lambda)$: distribution of λ ;
  ▷ k: number of iterations
  ▷ d: size of the dataset
  $\pi_{\text{piKL-IL}} \leftarrow \pi_{BC}$
  for $i \leftarrow 1, \ldots, k$ **do**
    Train a belief model $\hat{\mathcal{B}}$ from self-play games of $\pi_{\text{piKL-IL}}$
    Initialize dataset $\mathcal{D} = \emptyset$
    while len($\mathcal{D}$) $< d$ **do**
      Sample $\lambda_i \sim P(\lambda)$ for every player independently
      Generate a game g where player i follows piKL-LBS(λ_i, $\pi_{\text{piKL-IL}}$, $\hat{\mathcal{B}}$, π_{BC})
      Add the game g to dataset $\mathcal{D}$
    end while
    Train a new policy π' with behavior cloning on $\mathcal{D}$
    $\pi_{\text{piKL-IL}} \leftarrow \pi'$
  end for
  return $\pi_{\text{piKL-IL}}$
end procedure

It first trains an imitation policy π_{BC} via behavioral cloning on a dataset collected from the population of humans we want to model. Then piKL-IL iteratively improves a policy $\pi_{piKL-IL}$, alternating between generating higher quality data with piKL-LBS and training a better model using the generated dataset to produce a new $\pi_{piKL-IL}$.
piKL has great accuracy (sometimes better than human), depending on λ. So a distribution of λs is used to generate a spectrum of policies.
For two players, we run single-agent piKL-LBS with learned beliefs independently for both players, or run on one player and use imitation learning.

4.3 piKl-BR

We train a policy $\pi_{piKL-BR}$ to be a best response to $\pi_{piKL-IL}$ via Q-learning, but we modify the Q-learning update as:

$$Q(\tau_t^i, a_t) \leftarrow r_t(\tau_t, a) + \gamma \cdot Q(\tau_{t+1}^i, a'_{t+1})$$
$$\text{where } a'_{t+1} = \underset{a}{argmax} \left[Q(\tau_{t+1}^i, a) + \lambda \cdot \log \pi_{BC}(\tau_{t+1}^i, a) \right]$$

the exploration is ϵ-Greedy$\left[Q(\tau_{t+1}^i, a) + \lambda \cdot \log \pi_{BC}(\tau_{t+1}^i, a) \right]$.
When piKL-BR recieves an out of distribution input, it can produce bad results. This can be solved by returning piKL-LBS algorithm on top of it at test time, for producing π_{roll}

4.4 Experiments

2-Player Hanabi card game. (The deck consists of five color suits and each suit has ten cards divided into five ranks with three 1s, two 2s, two 3s, two 4s and one 5. At the beginning, each player draws

five cards from the shuffled deck. Players can see other players' cards but not their own. On each turn, the active player can either hint a color or rank to another player or play or discard a card. Hinting a color or rank, will inform the recipient which cards in their hand have that specific color/rank. Hinting costs an information token and the team starts with eight tokens. The goal of the team to play exactly one card of each rank 1 to 5 of each color suit, in increasing order of rank, The order of plays between different color suits does not matter. A successful play scores the team one point while a failed play one costs one life. If all three lives are lost, the team will get 0 in this game, losing all collected points. The maximum score is 25 points. The team regains a hint token when a card is discarded or when a suit is finished (playing all 5 of a suit successfully). The player draws a new card after a play or discard move. Once the deck is exhausted, the game terminates after each player makes one more final move.)

Human policy π_h is trained using 240k 2-player games. After training, the converged π_h gets 19.72 $\pm$ 0.10 in self-play and 63.63% in prediction accuracy on the test set. In Hanabi, the belief model takes the same AOH τ_i as the policy and returns a distribution over player i's own hand. The hand consists of 5 cards and we can predict them sequentially from the oldest to the newest based on the time they are drawn. The belief network ϕ consists of an LSTM encoder to encode sequence of observations and an LSTM decoder for predicting cards autoregressively.

We set $P(\lambda)$ to be a uniform mixture of Gaussian distributions $N(\mu, \sigma^2)$ truncated at 0 and 2μ with (μ, σ) = (1, 1/4),(2, 2/4),(5, 5/4),(10, 10/4) and each Gaussian is sampled with equal probability. We generate $d = 250K$ games in each iteration to train the new policy. In every LBS step, we perform $M = 10K$ Monte Carlo rollouts evenly distributed over $|A|$ legal actions. We sample $M/|A|$ valid private hands from the belief model to reset the simulator for rollouts. To better imitate policies under different λs, we feed the μ of the Gaussian distribution from which the λ is sampled to the policy network. Clearly, piKL-IL performs significantly better than π_h and the score increases as regularization λ decreases.

The BR is trained under a standard distributed RL setup where many parallel workers generate data with cross-play between the training policy and the fixed IL policy. piKL-BR is better at collaborating with piKL-IL than piKL-IL itself and the gap shrinks as the regularization λ decreases.

We run piKL-LBS on the piKL-BR policy with high regularization $\lambda = 2$. Imitation learning is used. Finally, we train an unregularized $\lambda = 0$ best response to the vanilla behavioral clone policy π_{BC} as our baseline. This agent achieves 23.19 ± 0.03 in cross-play with π_{BC} in convergence.

4.5 Results

2 experiments:

1. Ad-hoc team play with a diverse group of players without any prior communication (zeroshot)

	w/ Human Experts	w/ BR-BC	w/ piKL3
All Testers (56)	14.54 ± 1.47	**16.73 ± 1.27**	**17.18 ± 1.28**
Newcomer (2)	0.00 ± 0.00	0.00 ± 0.00	**10.00 ± 7.07**
Beginner (17)	9.12 ± 2.65	**14.82 ± 2.42**	**14.47 ± 2.63**
Intermediate (23)	14.57 ± 2.27	**19.48 ± 1.64**	**18.52 ± 1.79**
Expert (14)	**23.14 ± 0.60**	16.93 ± 2.41	19.29 ± 2.14

Table 2: The performance of different groups of players partnering with a group of testers. *Testers* are recruited from diverse sources to represent the general population with different skill levels and different conventions in Hanabi. Each *tester* is matched with one of the available human experts, one BR-BC baseline agent and one piKL3 agent in random order. The bottom 4 rows shows the results for each subgroup of testers. The number in parentheses after the group name is the headcount of that group. Each cell contains mean ± standard error.

Both BR-BC and piKL3 outperformed human experts in this task, indicating that playing with a diverse range of players in the zero-shot ad-hoc setting is challenging for humans. AI agents generally worked better with non-expert players, while experts have a clear lead when collaborating with other experts. This is likely because experts' behaviors are more predictable and the community has converged to a few well-known convention sets that are easy to identify for the experts who follow them closely in forums and discussion channels.

2. A group of expert players play multiple games with piKL3 and the BR-BC baseline in alternating order to further differentiate the gap between them.

	w/ BR-BC	w/ piKL3
Experts	21.21 ± 0.59	**22.23 ± 0.52**
	22.52% ± 3.96%	**31.86% ± 4.38%**

Table 3: The performance of experts playing with BR-BC baseline and piKL3 in alternating order repeatedly for a maximum of 20 games in total. The cells contain mean ± standard error (top row) averaged over 111 (BR-BC) and 113 (piKL3) valid games, and percentage of perfect games (bottom row).

Although the improvement may seem small numerically, the mechanics of Hanabi makes it increasingly difficult to improve as the score gets closer to a perfect 25. piKL3 outperformed the BR-BC in terms of both average score and percentage of perfect games. piKL3 achideve more perfect games with the experts. Experienced human players are particularly excited about perfect games as they are often significantly harder than getting 23 or 24 in a given deck.