Validating a New Collective Intelligence Technology for Accurate Ranking Using Artificial Swarm Intelligence

Gregg Willcox[1(✉)], Louis Rosenberg[2], and Hans Schumann[3]

[1] Unanimous AI, Seattle, WA 98109, USA
gregg@unanimous.ai
[2] Unanimous AI, Pismo Beach, CA 93448, USA
[3] Unanimous AI, San Francisco, CA 94123, USA

Abstract. We introduce hyperswarm ranking, a new collective intelligence technology based on the biological principle of Swarm Intelligence, and show that it enables networked human groups to collaboratively rank independent sets of items with significantly higher accuracy than traditional methods. While prior approaches, such as survey-based Wisdom of Crowd (WoC) techniques, are effective at amplifying groupwise accuracy, we show that this new approach significantly outperforms on a series of general knowledge questions, producing rankings that are 8.1% more accurate than WoC ($p < 0.01$). This translates into an impressive 39.5% amplification of the traditional "WoC effect." Finally, we show that the use of hyperswarm ranking enables networked human groups to generate groupwise rankings much faster than other commercially available tools that leverage the accuracy benefits of Artificial Swarm Intelligence, cutting the time required of human participants by more than half.

Keywords: Collective Intelligence · Artificial Swarm Intelligence · Collaboration Tools · Groupwise Ranking · HyperSwarms

1 Introduction

For well over a century, Collective Intelligence (CI) researchers have shown that human groups can amplify the accuracy of groupwise forecasts, estimations and evaluations. This is most commonly achieved through statistical aggregation methods referred to as Wisdom of Crowds (WoC) [1–3]. In a typical WoC scenario, individual estimates are collected from participants in isolation, often through blind survey. These estimates are then aggregated mathematically into an *average forecast* that is generally more accurate than the raw estimates produced by most individuals in the group. In recent years, an innovative new method has been developed that is not based on aggregating data from isolated individuals, but instead turns human groups into real-time systems moderated by intelligence algorithms modeled on the principle of Swarm Intelligence.

Known as Artificial Swarm Intelligence (ASI) or simply "Human Swarming," this method was first introduced in 2015 [1] and has been shown through many studies to

K. Arai (Ed.): FTC 2023, LNNS 813, pp. 290–304, 2023.
https://doi.org/10.1007/978-3-031-47454-5_22

significantly amplify the accuracy of groupwise forecasts [4–9, 14]. For example, a study conducted at Stanford University School of Medicine tasked groups of radiologists with diagnosing pneumonia based on chest x-rays [11]. When forecasting together as real-time swarms, diagnostic errors were reduced by over 30% compared to WoC methods [10].

While prior studies have shown that commercial ASI platforms (such as Swarm® from Unanimous AI) can significantly amplify the performance of human groups in a wide range of tasks [5–10] from forecasting sporting events [8–10] to predicting sales volumes of new products [13], the present research focuses on collaborative ranking tasks in which networked human groups must rank sets of independent items and converge on the most accurate orderings they can agree upon.

Prior studies have also extended the principle of Artificial Swarm Intelligence to the concept of Hyperswarms, in which real-time distributed groups are split into a plurality of overlapping subgroups such that each participant can only observe the fraction of other participants in their unique subgroup. Because the views of participants overlap but are not identical, each user is provided with a unique stimulus regarding the views of other participants and yet interactions among group members can still propagate across the full population as they do in traditional swarming interfaces. Theoretical research has shown that hyperswarms could enable groups to reach better decisions as compared to swarms with fully connected visibility [15, 16], but to date no human studies have confirmed this theory.

In this paper, we introduce a new ASI technology called hyperswarm ranking. This technology enables networked human groups to collaboratively rank independent sets of items with significantly higher accuracy than survey-based WoC approaches. In addition, hyperswarm ranking is significantly faster at reaching collaborative results than prior swarm-based methods. And finally, a number of innovations have been developed that enable this new ranking technology to leverage the benefits of real-time Swarm Intelligence while also enabling the majority of participants to engage the system asynchronously. This is a notable breakthrough for ASI systems, as all previous methods that leverage the power of Artificial Swarm Intelligence have required that the full population of participants engage synchronously with at least a sub-population of other participants. This is a significant logistical constraint that has been resolved in collaborative ranking tasks, as will be described in Sect. 2. It is also the first known test of hyperswarm technology on human subjects in a rigorously controlled task.

2 Method

2.1 The HyperRank Testbed

To quantify the value of this new hyperswarm ranking technology in authentic ranking tasks, we developed the HyperRank Testbed, a prototype system that enables distributed groups of human users to connect to a central server using any standard phone or personal computer running a standard browser. The HyperRank system provides a unique graphical user interface for collaborative ranking and enables networked groups to collectively rank sets of items, as shown in Fig. 1 below. Each question asked in the HyperRank system is answered through a series of discrete rounds in which collaborative groups

rank and re-rank items. For the testing described herein, three rounds of groupwise ranking were used. For each question, users were first shown the question text and answer choices (in a randomized order) and were then asked to optimize the order for maximum accuracy by dragging selected items up or down the list. They completed this step (Round 1) without influence from any other participants.

When enough users have finished their rankings, a **Baseline Aggregated Ranking** was calculated across the pool of users who have submitted their individual rankings. In a typical ASI approach, this baseline ranking would be shared across the full population of participants as a "stimulus" to evoke behavioral feedback. This feedback would be achieved by asking the individuals to consider the baseline aggregated ranking and to collaboratively improve its accuracy by making real-time adjustments. That said, the HyperRank system does NOT share the baseline ranking with the group. That's because doing so would greatly limit the diversity of behavioral responses generated across the population of participants. After all, the full population would be reacting to the exact same baseline aggregated ranking.

Instead, a new method was developed in which a **Probabilistic Ranking Model** is computed from the baseline ranking data and is used to randomly generate unique probabilistic rankings to be shown to each participant. In this approach, the full set of unique probabilistic rankings are crafted algorithmically so they average out to the original baseline aggregated ranking. In this way, the population (when viewed as a whole) is stimulated with a distribution of rankings that together faithfully represent the baseline aggregated ranking. That said, when viewed on an individual level, the participants are each exposed to unique rankings. This effectively connects a group of users into a hyperswarm: each user observes and is asked to optimize a ranking that represents the combined rankings of only a few other users in the group. No user sees the full picture, but by connecting the group in this way, user sentiments propagate through the full population. Over a series of three rounds, the sentiments propagate through the population and the system amplifies the group's collective intelligence.

This method uses an innovative algorithm we call **Probabilistic Rank Aggregation**. It calculates a unique probabilistic ranking for each individual participant in the group based on the ranking response they submitted in the prior round in combination with (i) the Probabilistic Ranking Model generated for the prior round, and (ii) the individual rankings that have already been shown to users this new ranking round.

As we'll discuss in the sections that follow, this algorithm has three key features that contribute to the ability of the HyperRank system to amplify the collective accuracy of groupwise rankings: (i) it accurately samples the baseline rankings collected from the prior round, ensuring that the distribution of probabilistic rankings shared with participants averages out to the baseline aggregated ranking, (ii) it reduces noise in the aggregation process by clipping outliers from baseline ranking data, and (iii) it challenges each user by presenting them with a customized probabilistic ranking that is more likely to disagree with their own ranking and thus evoke an informative behavioral response. In this way, this unique method elicits more diverse behavioral data from the group in each round without compromising the overall collective intelligence of the group's prior rankings. We elaborate on the Probabilistic Aggregation algorithm in Sect. 2.2.

Once a unique probabilistic ranking is calculated for a given user using Probabilistic Rank Aggregation, an animated robot arm appears on their screen and modifies that user's submitted ranking (from the prior round), visually reordering their ranking into the unique probabilistic ranking that has been assigned to that user. Users are told that the aggregated ranking on their screen was generated algorithmically from a set of other users' rankings in the previous round. The user is then tasked (in this new round) with adjusting the provided ranking to maximize accuracy by moving items up or down as they see fit. That said, there's an important restriction – each user is only allocated a limited number of "moves" to adjust the ranking they are shown. The number of moves allocated to each user is limited to half of the moves that were required for the animated robot arm to adjust their prior ranking to the unique probabilistic ranking, rounded up. In this process, a "move" is counted as a single action of dragging an item up or down the ordered list from one position to any other. For example, moving an item from rank 7 to rank 2 is one move, as is moving it from rank 5 to rank 6.

The limitation of moves accomplishes two things: first, it means that users cannot simply revert their ranking back to their last-round ranking. Instead, they must prioritize which adjustments (i.e. moves) they believe will best optimize the accuracy of the ordering. This *forced prioritization* reveals conviction information to the HyperRank system for each individual user. After all, each user is driven to reveal their highest-conviction rankings based on how they use their limited supply of moves. And because "a move" allows a user to reorder an item as far as they deem necessary up or down the list, users often prioritize movements that are the furthest from the location they believe would maximize accuracy. This tends to favor moves towards the extremes of the list (the top and bottom), revealing the items that have the highest or lowest confidence in satisfying the question asked. This is helpful, as ranked lists are generally used most by human groups to identify the topmost and bottommost items, with less importance often given to items in the middle.

It's important to note that users are not required to use all their moves in a given round, and when they choose not to use all their moves, this too reveals important conviction information to the HyperRank system. That's because a decision by a user not to use all their available moves reveals their ambivalence between the ranking of the unmoved items in the provided probabilistic ranking and their own prior ranking (i.e. they have chosen NOT to return some items to the position they originally put those items in in despite having available moves to do so). When this happens, the HyperRank system algorithmically infers low conviction in their prior ranking of these items.

After a user has finished adjusting the provided probabilistic ranking, the process repeats for Round 3. Once again, each user is provided with a unique ranking (this time based on the collected responses for Round 2) and once again each user is given a limited number of moves to optimize the provided ranking. This process again evokes powerful behavioral information from each user, indicating their varying levels of confidence and conviction with respect to the placement of various items in the list.

Finally, the confidence and conviction information (across all rounds) is used to compute a **Final Group Ranking** based on the unique behavior of all users across the full process. This unique method is detailed in Sect. 2.3.

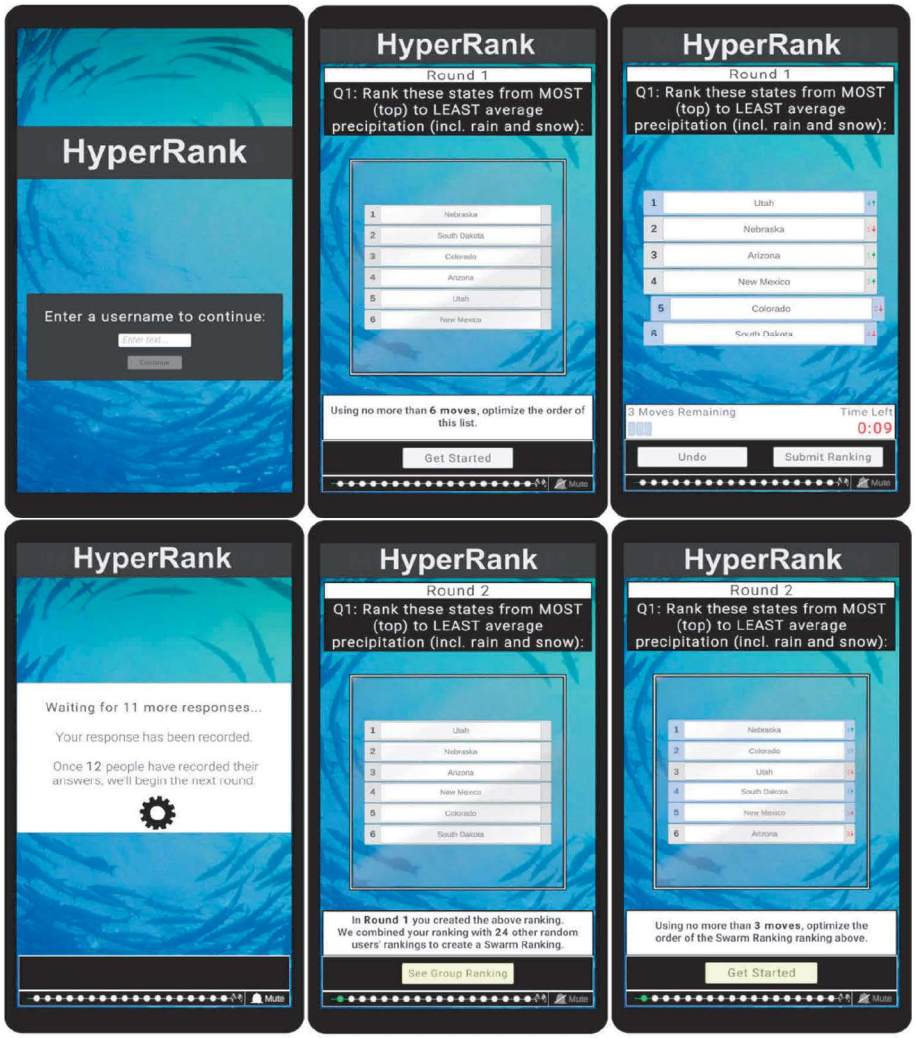

Fig. 1. Views of a user proceeding through the first question in a HyperRank session. From top-left to top-right, then bottom-left to bottom-right: (i) the user logs into the platform with a username, (ii) the user is presented with a question and a list of items and is asked to order the list (iii) the user orders those items in 45 s or less and submits their ranking. (iv) when enough users have submitted their answers, the next round begins. (v) each user is told their ranking was combined with some number of other user's rankings, and (vi) is then presented a new ranking that was created by the group, which they must optimize using only a handful of moves. Steps (iii) to (vi) repeat for round 2, and only step (iii) repeats for round 3.

2.2 Probabilistic Rank Aggregation

The unique collective intelligence method described herein requires that during each new round of groupwise ranking, the platform generates a probabilistic distribution of aggregated rankings (based on prior round data) and sends a **Unique Probabilistic Ranking** to each user for consideration in the new round. To achieve this, the following steps are performed: (i) calculate an **Aggregation Matrix** from the response data received so far, (ii) generate a candidate ranking from this Aggregation Matrix using the Probabilistic Aggregation approach outlined in below subsection, (iii) generate a Unique Probabilistic Ranking by repeating step (ii) to generate a set of candidate rankings and selecting the candidate ranking that is the furthest distance from the targeted user's previously submitted ranking. Each of these steps are described in detail in the sections that follow. A high-level flowchart of this process is shown in Fig. 2.

Round 1 Received Rankings

	User 1	User 2	User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10
1st	Apple	Apple	Apple	Apple	Apple	Apple	Apple	Apple	Apple	Apple
2nd	Cherry	Banana	Banana	Cherry	Banana	Cherry	Banana	Banana	Banana	Banana
3rd	Banana	Durian	Cherry	Durian	Cherry	Banana	Cherry	Cherry	Durian	Durian
4th	Durian	Cherry	Durian	Banana	Durian	Durian	Durian	Durian	Cherry	Cherry

Round 1 Aggregation Matrix

Ranking	Apple	Banana	Cherry	Durian
1st	100%	0%	0%	0%
2nd	0%	70%	30%	0%
3rd	0%	20%	40%	40%
4th	0%	10%	30%	60%

Round 2 Unique Probabilistic Rankings

	User 1	User 2	User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10
1st	Apple	Apple	Apple	Apple	Apple	Apple	Apple	Apple	Apple	Apple
2nd	Banana	Cherry	Banana	Banana	Cherry	Banana	Cherry	Banana	Banana	Banana
3rd	Cherry	Banana	Durian	Durian	Durian	Cherry	Banana	Durian	Cherry	Cherry
4th	Durian	Durian	Cherry	Cherry	Banana	Durian	Durian	Cherry	Durian	Durian

Fig. 2. The diagram above shows the Probabilistic Aggregation Model generating Unique Probabilistic Rankings for 10 users in a HyperRank session from the Received Rankings of those users. First, the Received Rankings are converted into an Aggregation Matrix. Then, Unique Probabilistic Rankings are generated for each user from this Aggregation Matrix. There are two important things to note about the Unique Probabilistic Rankings generated here: first, each user received a different ranking than the one they submitted, and second, the Unique Probabilistic Rankings have the same group-level overall mean ranking and ranking distribution as the Aggregation Matrix.

Calculation of the Aggregation Matrix. For each round, an Aggregation Matrix (E) is calculated for the submitted answers that describe the degree of confidence the group

collectively expressed in support of each item being placed at each rank. The matrix has N rows and N columns (where N is the number of items being ranked), and each cell in the matrix contains a single number. The higher the number in the cell, the greater is the group's collective conviction that the item should be ranked in that location. Each row in this matrix represents a given rank (e.g. 1st) and each column represents a given item (e.g. "Apple"). So, element E(3,2) represents the conviction with which Banana (item 2) is ranked 3rd.

In the first round this number represents the simple fraction of users that responded with that answer in that rank, but in future rounds other behavioral information is taken into account to weight some users higher or lower than others for particular items in particular ranks based on the conviction detected or inferred for each user when ranking that item in that location. This behavioral information includes, for example, the prior round ranking data collected for each user, the adjustments that user made to the group ranking in the current round, and the number of moves that user had left (i.e. chose not to use) at the completion of that round.

An example Round 1 aggregation matrix is shown in the second step of Fig. 2. In this example, the group was confident that Apple should be ranked first: 100% of users ranked Apple in 1st place, and 0% of users gave any other answer for first place. On the other hand, the group was divided on the best location for Cherry, with 30% of users ranking Cherry in 2nd place, 40% of users ranking Cherry in 3rd place, and 30% of users ranking Cherry in 4th place. In future rounds these values would also reflect the relative conviction expressed by users for particular items ranked in particular locations. In this way, the Aggregation Matrix is a probabilistic representation of the aggregated weighted opinions of the population of users.

Probabilistic Aggregation. Now that we have an Aggregation Matrix that represents the group's aggregated opinions, we can use this matrix to generate a distribution of unique individual rankings to be sent to the population of users, this distribution being probabilistically equivalent (as a set) to the aggregated collective ranking expressed by the group. For each individual user, this is done by generating a candidate ranking to show to that user. It is performed in three steps: first removing outliers, then calculating the probability that each item should be selected in each rank, and then selecting the items from this matrix to fill each rank in the unique ranking generated for that user.

First, outliers in the data are dynamically removed from the group's Aggregation Matrix. This process identifies data lying on the edges of each item's range of rankings and moves these outliers closer to the center of the rankings. In current implementation, "outliers" are defined as any ranking that is greater than the 90th percentile of high or low rankings of the item. When outliers are detected, their rankings are set to the 90th percentile. For example, consider a set of 12 responses: 10 out of 12 users ranked "Apple" between the 2nd and 4th ranks, one user ranked Apple 1st and one user ranked Apple 6th. The Aggregation Matrix would show that the 1st and 6th-place rank frequency entries for Apple are respectively in the 91st percentile of high and low rankings for item Apple. The values for these entries would be added to the entries for 2nd and 4th places respectively and then set to 0. This mechanism cleans the Aggregation Matrix of outliers by scaling their responses away from extremes, while still recognizing the

ordinal nature of rankings (i.e., outliers are not simply removed, they are instead moved inward towards the mean until they no longer fall outside the 90th percentile).

Next, a Selection Probability Matrix (**S**) is calculated from this cleaned Aggregation Matrix that indicates the rough probability with which each item should be shown in each rank to a new user in the new round. Feedback is used to ensure that the frequency with which items are shown in each rank is close to the frequency with which they were received in that rank. This feedback algorithm works as follows: it takes as input three NxN-shaped Rank-Frequency Matrices: the Aggregation Matrix collected from users so far in the previous round (**E**), a Aggregation Matrix of the rankings presented to users already for the new round (**H**), and the latest Aggregation Matrix of the user for whom this unique aggregation is being generated (**U**).

$$F = Normalize(Max(E - H, 0)) \tag{1}$$

$$S = Normalize(Max(F - U/c, 0) + epsilon * E) \tag{2}$$

Equation (1) calculates the feedback term to show rank-items in a way that approaches the received distribution. Equation 2 reduces the likelihood of giving the user rankings that match their submitted rankings, or order to promote each user's rankings being challenged by the group's rankings. The constant **c** can be set to any number: larger numbers reduce the frequency that users are presented with rankings that challenge their own, and smaller numbers increase this frequency. In practice, **c** is set to the number of responses received to this round so far, so that the "**-U/c**" term explicitly cancels this user's own contribution to the Aggregation Matrix. The "**epsilon*E**" term ensures that there's a small but nonzero chance of finding each item in each ranking in which it was received. This helps prevent issues that arise when selecting items in the next step. Epsilon is a tiny, positive, nonzero value. The final Selection Probability matrix entries are clamped to be greater than 0 and are normalized so that they sum to 1.

Finally, the selection probability matrix is used to calculate a list of items to show the end user. Importantly, items can be selected only once, and each item must be present in the final list: in this way, only the order of the list changes between rounds, rather than the content of the list.

To ensure that the topmost and bottommost ranks are selected with maximum fidelity to the selection matrix, items are selected from most extreme ranks to least extreme ranks, starting from the top. This is done by alternately iterating down the ranks, starting from rank 1 and then rank n, and working towards the center of the list. For example, with 7 items, the ranks considered in order would be: [1–7].

For a given rank **n**, the probability of selecting each remaining item is calculated as the frequency of that item in the nth position and all positions more extreme than **n**. In the example above, for rank 3, the probabilities of finding each item in ranks 1, 2, and 3 are summed up and normalized. One item is selected from the remaining set of items with these normalized probabilities. Finally, this item is removed from the list of remaining items and the process is repeated until no items (and therefore no ranks) remain.

Final Ranking Selection. Now that we have an algorithm for probabilistically generating rankings with high fidelity, we need to use it to create a Unique Probabilistic Ranking

for the user. While it's possible to use the ranking that is generated from Probabilistic Aggregation as the Final Ranking, the generated ranking might be identical to the user's ranking from the previous round due to the probabilistic nature of the algorithm. To ensure that the ranking shown to the user challenges their last-round ranking, multiple candidate rankings are generated using Probabilistic Aggregation and the ranking that challenges the user the most is selected as the Unique Probabilistic Ranking to be shown to the user. The degree to which a ranking 'challenges' the user is calculated as the distance between the user's last-round ranking and each generated ranking. This distance is measured as a combination of: (i) the number of moves that would be required to change one ranking into the other ranking, (ii) the average move distance of those moves, and (iii) the average distance of moves from the edges of the list. In this study, 5 candidate rankings were generated, and the Unique Aggregated Ranking shown to each user was selected in this way.

2.3 Calculating the Collective Ranking

The output of the HyperRank system is a collective ranking that represents the ranking the group could best agree upon. This collective ranking is calculated for each round as the argument-sorting over each item **n** of the rank-weighted sum of the Aggregation Matrix for that round, as shown in Eq. (3).

$$Ranking = ArgSort_i(\sum\nolimits_n E_u(n, i) * n) \tag{3}$$

The Round 1 final collective ranking is equivalent to a WoC ranking, because users were given enough moves to create whatever ranking they wanted and created their ranking without any influence from other users or the HyperRank system.

2.4 Study

To quantify the effectiveness of the unique collaborative ranking methods described above, a formal set of human tests were conducted using human subjects, each of whom was sourced through the Amazon Mechanical Turk service and paid a small participation fee for their time.. These participants were divided into six groups of approximately 25 members, each group assigned to one of six different experimental sessions. The users assigned to each session logged into a dedicated software platform at an assigned time for that session. Each session lasted approximately 15 min.

In each of the six experimental sessions, the group of approximately 25 members was tasked answering a set of six general-knowledge ranking questions. While the six questions used in each of the six different experimental sessions were not identical, they followed a very similar structure and format. Specifically, there were four types of question used in this study: *"Rank these US states by average precipitation (in inches, including rain and snow)"*, *"Rank these US states by fraction of democrat voters"*, *"Rank these US states by average temperature"*, and *"Rank these countries by Gross Domestic Product"*. Each question contained between six and nine items to be ranked. No two questions contained same set of items, and no question was repeated.

Upon completion of the six experimental sessions, each with six ranking questions, a total of 36 unique ranking tasks were conducted. All participants were instructed to answer the questions to the best of their ability but not to look up the answers. Sessions lasted between 12 and 16 min each. Users were paid the same amount for their time regardless of their performance, so there was no motivation for cheating. In addition, the tasks were performed under significant time pressure (only 45 s per round) so there was not sufficient time to cheat by looking up answers.

2.5 Grading

The correct ranking $\mathbf{R_{correct}}$ for each question was calculated from the data and was used to calculate the Rank Accuracy of each ranking $\mathbf{R_i}$. The Rank Quality, as shown in Eq. 4, is a score between 0% and 100% that measures the Root Mean Squared Error (RMSE) of the ranking compared to the worst and best possible rankings: [Eq. 4]. $\mathbf{R_i}$ in this equation represents a ranked list created by an individual or the group consisting of a set of answers N, where $\mathbf{R_i(N)}$ is the ordinal ranking of answer **N**. For example, a ranking: [1st: Apple, 2nd: Banana, 3rd: Cherry] would be represented as $\mathbf{R_i}$ = [Apple: 1, Banana: 2, Cherry: 3], and R_i(Banana) = 2. $\mathbf{R_{correct}}$ is the correct ordinal ranking to each question, and $\mathbf{R_{worst}}$ is the worst ordinal ranking (i.e. the reverse of $R_{correct}$).

$$RMSE(R_i) = RMSE(R_i, R_{correct}) = \frac{1}{n}\sqrt{\sum\nolimits_n [R_i(n) - R_{correct}(n)]^2} \tag{4}$$

$$RankQuality(R_i) = \frac{RMSE(R_i) - RMSE(R_{worst})}{RMSE(R_{correct}) - RMSE(R_{worst})} \tag{5}$$

This metric was chosen for two reasons: first, it penalizes answers that are farther away from the correct location, and second, it allows comparisons across questions with different numbers of items. Without this normalization to a [0, 1] scale, questions with more items would have a higher average RMSE simply due to the number of items and possibility for items to be ranked further from their correct locations.

The average Rank Accuracy of individual rankings made in the first round was also calculated using this same approach to quantify the quality of rankings created by individuals in isolation. To do so, the Rank Accuracy of each ranking made by each individual was tallied and then averaged.

Finally, three subsets of each ranking were graded using this approach: the full ranking, the top answer, and the bottom answer. To calculate the Rank Accuracy of the top and bottom answers, the ranked lists were subset to only the top and bottom answers respectively, and then the Rank Accuracy of only those 1-item lists were calculated. A score of 100% for the Top metric, for example, indicates that the ranking's top answer was correct, while a ranking in which the top answer was actually the 3rd-most correct answer in the correct list out of a total of 9 items would receive a score of 75%.

3 Results

The group's collective ranking after each of the three rounds was calculated and compared to the Average Individual Rank Accuracy. Figure 3 shows the Rank Accuracy of the average individual (before aggregation) and then of the group's collective rankings

after each round, where Round 1 is equivalent to a WoC aggregation. The average individual scored a Rank Accuracy of 47.6% in this question set, while simply collecting a set of Round-1 rankings (WoC) improved this to a 59.5% Rank Accuracy. Over each subsequent round in HyperRank, the Rank Accuracy further improved, leading to a 64.2% Rank Accuracy by the end of Round 3. As shown in Table 1, HyperRank's improvement over the WoC effect was highly statistically significant ($p < 0.01$), as measured using a paired t-test over the full sample of 36 questions. We can therefore conclude that the HyperRank system allowed this group to create more accurate rankings to these questions in a way not explainable by random chance alone.

Next, we examined the Rank Accuracy of the top- and bottom-ranked items created in each aggregation. Again, each subsequent round yielded an increasing Rank Accuracy: the top answer from the round 3 HyperRank aggregation (84.1%) significantly outperformed the top answer from the round 1 WoC aggregation (77.7%, $p < 0.05$), and the bottom answer from round 3 (82.4%) outperformed the round 1 WoC aggregation (77.5%, $p = 0.065$), though this effect was not significant at the 5% alpha level.

As compared to the traditional WoC effect, which showed a 11.9% improvement over the average individual's full rankings on this question set, the HyperRank system improved over the individuals by a further 4.7% (for a total improvement of 16.6%), equivalent to amplifying the WoC effect by 39.5%.

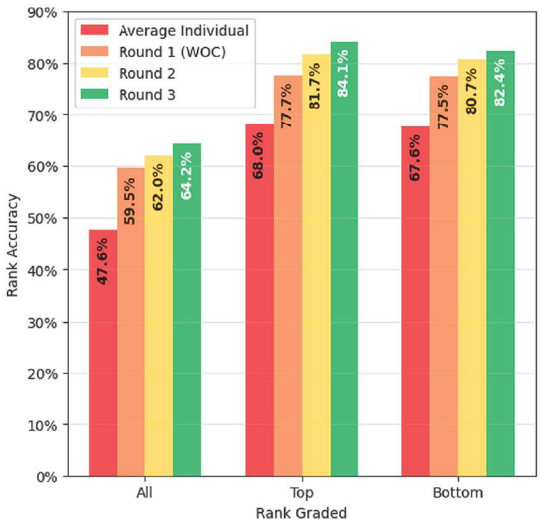

Fig. 3. Bar plot showing the rank accuracy after each round for each different rank graded.

Table 2 details the statistical comparisons between HyperRank's final rankings and the Average Individual rankings across all metrics. Each metric (all of the ranks, the top-ranked item, and the bottom-ranked item) was significantly more accurate when using HyperRank as compared to the average individual ($p < 0.001$ in all cases), providing strong evidence that HyperRank produces better rankings than an average individual.

Table 1. Rank accuracy of wisdom of the crowd compared to HyperRank.

Ranks graded	Wisdom of Crowd Rank Accuracy	HyperRank Rank Accuracy	% Increase in Rank Accuracy	p-value
All	59.5%	64.2%	8.1%	0.002
Top	77.7%	84.1%	8.2%	0.033
Bottom	77.5%	82.4%	6.3%	0.065

Table 2. Significance test results against average individual for different ranks graded.

Rank graded	Average Individual Rank Accuracy	HyperRank Rank Accuracy	% Increase in Rank Accuracy	p-value
All	47.6%	64.2%	34.9%	<0.001
Top	68.0%	84.1%	23.7%	<0.001
Bottom	67.6%	82.4%	21.9%	<0.001

3.1 Rank Improvement by Question

In the beginning of Sect. 3, we found that HyperRank improves the average Rank Accuracy for the three ranking subsets graded. To better understand whether this improvement was due to a large improvement over just a few questions or a broader improvement across many questions, we analyzed how HyperRank changed the Rank Accuracy of each question in the question set.

Table 3 shows the change in Rank Accuracy scores from the WoC to HyperRank, as categorized by number of questions in which the Rank Accuracy scores improved (i.e. HyperRank outperformed WoC), stayed the same, or worsened. When all ranks were graded, HyperRank's Rank Accuracy improved over the WoC ranking 20 times out of 36 (56%), made no change 10 times (28%), and worsened only 6 times (17%). Using a Sign Test, we found HyperRank was significantly more likely to improve the Rank Accuracy over the WoC than it was to make it worse for all ranks (fraction improved = 20/26 = 76.9%, $p = 0.005$) on a single question.

For the Top (fraction improved = 4/4 = 100.0%, $p = 0.063$) and Bottom (fraction improved = 6/8 = 75.0%, $p = 0.145$) ranks, insignificant evidence was found to show HyperRank improved the Rank Accuracy over the WoC, in part due to a low frequency of cases in which an improvement can even be made.

Table 3. Change in rank accuracy from WoC to HyperRank, summarized by number of questions in each category.

Rank graded	Increased	Stayed the Same	Decreased	Sign Test p-value
All	20	10	6	0.005
Top	4	32	0	0.063
Bottom	6	28	2	0.145

3.2 Time Analysis

Each session in this study required between 12 and 16 min to complete, and on average took 13.8 min to complete, equivalent to 2.3 min per question and 46 s per round. Swarm, another collective intelligence interface that uses ASI technology, takes about 1 min per question, but would use N−1 questions to rank N items using a procedure of elimination. There were on average 7.4 items in each question in this study, for an average time estimate of 6.4 min per question using the Swarm platform. As a result, HyperRank generated collective rankings 64% quicker than the best alternative ASI tool: Swarm.

4 Conclusions

In this work we outlined a proof-of-concept system for collective ranking using the principles of Artificial Swarm Intelligence and Hyperswarms, called HyperRank. We conducted the first study of a hyperswarm system with real human users (to the author's knowledge) and demonstrated that HyperRank enables groups of users to generate collective rankings quickly and with a high degree of accuracy on a set of general-knowledge questions. The full collective rankings that were generated with HyperRank significantly outperformed both the average individual ($p < 0.001$) and the Wisdom of the Crowd ($p < 0.01$), and the Wisdom of the Crowd effect was amplified by over 40% when using HyperRank. We further demonstrated that these effects held for not only the full rankings, but also for the top- and bottom-ranked items, indicating that HyperRank could be used to accurately estimate the top and bottom-ranked items as well as to generate full rankings of items. Finally, we showed that this approach to collective ranking is 64% faster than the current state of the art ASI application, making it easier to use in practice.

Interestingly, the accuracy of the collective rankings increased through each round; would the groups have reached even better answers if given more rounds for each question? After what number of rounds does the marginal accuracy tend to 0, and how can administrators of these sessions make tradeoffs between this marginal accuracy gain and the marginal time taken for each round? Future work may run sessions with more rounds to investigate.

While this study showed promising initial results, it barely scratched the surface of the behavioral data collected in this platform and doesn't fully explain why or how the system works: future studies should address this. Future work will apply this system to other question types, including forecasting and sentiment (e.g. in market research).

Another large open question is synchronicity: the current experiment was conducted using synchronous groups, but the platform has been built to enable groups to asynchronously create group rankings. Asynchronous capability would help to dramatically reduce logistical barriers to using ASI tools, so future experiments should explore this capability and both time savings and whether asynchronous collaboration yields similar accuracy improvements over the WoC. Finally, future work could also compare to Delphi or other round-based aggregation techniques.

Acknowledgments. The authors would like to thank Patty Sullivan and Chris Hornbostel for their assistance in collecting data for this study, and Unanimous AI for the use of the HyperRank platform.

References

1. Rosenberg, L.: Human Swarms, a real-time method for collective intelligence. In: Proceedings of the European Conference on Artificial Life, pp. 658–659 (2015)
2. Galton, F.: Vox populi. Nature **75**, 450–451 (1907)
3. Steyvers, M., Lee, M.D., Miller, B., Hemmer, P.: The wisdom of crowds in the recollection of order information. In: Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C.K.I. (eds.) Advances in Neural Information Processing Systems (2009)
4. Rosenberg, L.B.: Human swarms, a real-time method for collective intelligence. In: Proceedings of the European Conference on Artificial Life, pp. 658–659 (2015)
5. Rosenberg, L.: Artificial swarm intelligence vs human experts. In: 2016 International Joint Conference on Neural Networks (IJCNN), pp. 2547–2551. IEEE (2016)
6. Rosenberg, L., Baltaxe, D., Pescetelli, N.: Crowds vs swarms, a comparison of intelligence. In: IEEE 2016 Swarm/Human Blended Intelligence (SHBI), Cleveland, OH, pp. 1–4 (2016)
7. Baltaxe, D., Rosenberg, L., Pescetelli, N.: Amplifying prediction accuracy using human swarms. In: Collective Intelligence, New York, NY (2017)
8. Willcox, G., Rosenberg, L., Askay, D., Metcalf, L., Harris, E., Domnauer, C.: Artificial swarming shown to amplify accuracy of group decisions in subjective judgment tasks. In: Arai, K., Bhatia, R. (eds.) FICC 2019. LNNS, vol. 70, pp. 373–383. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-12385-7_29
9. Rosenberg, L., Pescetelli, N., Willcox, G.: Artificial swarm intelligence amplifies accuracy when predicting financial markets. In: 2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON), New York City, NY, pp. 58–62 (2017)
10. Rosenberg, L., Willcox, G.: Artificial swarm intelligence vs Vegas betting markets. In: 2018 11th International Conference on Developments in eSystems Engineering (DeSE), Cambridge, United Kingdom, pp. 36–39 (2018)
11. Rosenberg, L., Lungren, M., Halabi, S., Willcox, G., Baltaxe, D., Lyons, M.: Artificial swarm intelligence employed to amplify diagnostic accuracy in radiology. In: 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, pp. 1186–1191 (2018)
12. Rosenberg, L., Willcox, G.: Artificial swarms find social optima: (late breaking report). In: 2018 IEEE Conference on Cognitive and Computational Aspects of Situation Management (CogSIMA), pp. 174–178 (2018)
13. Willcox, G., Rosenberg, L., Schumann, H.: Sales forecasting, polls vs swarms, 14 March 2019. https://doi.org/10.2139/ssrn.3390043, Accessed 16 Jan 2020

14. Rosenberg, L., Willcox, G.: Artificial Swarm Intelligence (2019). Unanimous AI. https://unanimous.ai/whitepaper, Accessed 11 May 2023
15. Rosenberg, L., Domnauer, C., Willcox, G., Schumann, H.: From swarms to hyperswarms: a new methodology for amplifying group intelligence. In: Arai, K. (ed.) FTC 2021. LNNS, vol. 358, pp. 239–251. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-89906-6_17
16. Willcox, G., Rosenberg, L., Domnauer, C., Schumann, H.: Hyperswarms: a new architecture for amplifying collective intelligence. In: 2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, pp. 0858–0864 (2021). https://doi.org/10.1109/IEMCON53756.2021.9623239