

8 Appendix

8.1 Concordia Contest Details

The 2024 NeurIPS Concordia Contest attracted 197 individual participants who together made 878 submission attempts, with 25 teams ultimately submitting their final agents for evaluation. Entrants were tasked with designing a single language-model-powered agent to compete across five cooperation-eliciting scenarios—Pub Coordination, Hagglng, State Formation, Labor Collective Action, and Reality Show—that each probe facets of social intelligence such as promise-keeping, negotiation, reciprocity, reputation management, partner choice, compromise, and sanctioning. Under the hood, every agent interacts with the environment via natural-language observations and action intents, all mediated by a Game Master which resolves those intents into world events. Agents are evaluated in both self-play and cross-play modes, and their final ranking is determined by the average returns across scenarios. The contest unfolded in two main phases: a Development Phase from September 15th to November 15th, 2024; an Evaluation Phase beginning immediately afterward, with final submissions due November 19th and winners announced at the NeurIPS 2024 contest session in December.

Concordia agents have both long-term memory and working memory, allowing them to maintain coherent identities and behaviors over time. This architecture enables agents to exhibit contextually appropriate behavior informed by social norms, personal history, and situational understanding—capabilities essential for navigating mixed-motive social interactions. Agents receive natural language descriptions of their local environment and can take arbitrary actions by generating unstructured natural language outputs. The Game Master then determines the effects of these actions based on the current state of the simulation.

Concordia’s environments are richly narrative and open-ended. To ensure contextual depth, each agent-instance is initialized with a unique life history—memories, personality traits, and social station—that remains hidden from the decision process (“veil of ignorance”). This design requires participants to craft an agent architecture capable of generalizing across a diverse population of individuals, rather than tailoring behavior to any single backstory.

Contest Structure Contestants submitted agents that were run in a series of environments. These environments were designed to be ‘cooperation-eliciting’ within their respective contexts. To perform well as individuals, agents needed to cooperate skillfully, which enabled the assignment of cooperation scores based on individual returns. Although short-term incentives existed for agents to defect from cooperative play, such actions typically led to lower social welfare and reduced individual performance. Each environment was equipped with a language model-based reward model that assigned quantitative scores to agents based on the outcomes generated by the game master (GM), with scoring being highly contextual to each environment.

The evaluation phase consisted of two sub-phases. First, each submitted agent was evaluated across multiple varied environments, resulting in an Elo score and a corresponding agent rank. Second, the top six ranked agents were re-evaluated in a tournament-style phase to determine the final cooperative ranking. Elo scores were recomputed for these six agents, and this final ranking was used to inform the overall score.

Several language models were evaluated on the basis of performance, affordability, and accessibility. Gemma-2-9b-it was selected as the official LLM for the competition. Most other language models were available to users for their own development and iteration process.

The contest was administered and managed using the Codabench platform². A slack group was made available for contest participants to coordinate with one another and engage with the contest hosts.

8.1.1 Contest Rules

1. **Agent Implementation:** Participants have the liberty to design, train, and implement their agents using any approach they deem fit. However, it is imperative that during the evaluation phase, agents operate autonomously without seeking external assistance. This includes, but is not limited to, prohibiting the use of plug-ins, APIs, or accessing external databases and

²Codabench site for Concordia: <https://www.codabench.org/competitions/3888/>

information resources not explicitly provided or permitted within the contest framework. The intention is to ensure that all agents rely solely on their capabilities and the resources made available through the contest to perform tasks and make decisions.

2. **Competition Structure:** The contest is segmented into two main phases: the development phase and the evaluation phase. During the development phase, participants can submit their agents for evaluation once every 24 hours, receiving feedback on their performance via an automated score. Although these submissions impact the ongoing leaderboard, they do not count towards final rankings. During evaluation, participants will be notified of their scores post-submission, with full rankings disclosed at the contest’s conclusion.
3. **Limitation on LLM Calls:** There will be a strict limitation on the number of Large Language Model (LLM) calls an agent can make per step. This policy serves two primary purposes: first, to maintain a level playing field by ensuring all participants’ agents are within the same "weight category," minimizing the advantage that could be gained from access to superior computational resources. Second, it provides a predictable upper bound on evaluation time and associated costs, making the contest more manageable and accessible. This ensures that the creativity and strategic input of each participant are central to the competition, within the bounds of equitable computational use.
4. **Source Code Submission:** While releasing source code is not a prerequisite for leaderboard acknowledgment, the contest reserves the right to withhold prizes from entries not disclosing their source code. All submissions in the evaluation phase must, however, privately share their source code with organizers for verification and adjudication purposes.
5. **Singleton Entries:** Multiple entries by single participants or collaborative entries that significantly overlap will be disqualified. Participants must contribute to only one team.

8.2 Technical Implementation

The Concordia Contest uses a consistent technical infrastructure to ensure fair and reproducible evaluation. All agents interact with the environment through the Concordia API, which provides a standardized interface for observation and action generation. To ensure accessibility and fairness, the contest limits the computational resources available to each agent, including the number of LLM calls per round and the size of input and output tokens. A Google Colab was provided as a no-code/low-code agent design mechanism.³ The Contest was run on the Codabench platform, which managed submissions, announcements, and the contest leaderboard.⁴

Ranking of Agents We present the full rankings of all agent submissions under multiple evaluation methodologies. These include Elo scores (Table 2), three voting-based methods—Iterative Maximal Lotteries, Copeland, and Ranked Pairs Tables 3 to 5—as well as Evaluation without Aggregation (EwA), shown in Table 5. This experimental method does not leverage statistics (e.g., no mean/average) to reduce the multiple runs per task, instead introduces a multi-player ranked-ordered tournament (i.e., team chess) to reduce the task-runs axis.

These tables allow us to understand how agent rankings vary under different assumptions about outcome aggregation and strategic interaction. While Elo remains the primary metric for its interpretability and consistency with standard reinforcement learning setups, voting-based methods and EwA provide valuable alternative perspectives—particularly in settings where agent strengths are non-transitive or results are sparse. Note that across all tables, the agent `taehun_gcal`, ranked 4 with Elo, that ended up winning the final six, was never ranked among the top-five submissions during evaluation.

Final Five Crossplay. To assess robustness in cross-play among the top agents, we re-evaluated the final five using methods from Voting as Evaluation [34], implemented via OpenSpiel [36]. Specifically, we applied Iterative Maximal Lotteries (table 8), Copeland (table 9), and Ranked Pairs (table 10) voting rules to aggregate pairwise outcomes into global rankings. These methods capture robustness to cyclic dominance and pairwise inconsistencies, offering a complementary perspective

³Google Colab is available at https://colab.research.google.com/github/google-deepmind/concordia/blob/main/examples/three_key_questions.ipynb

⁴The codabench platform is available here: <https://www.codabench.org/competitions/3888/>

Rank	Submission	Elo
1	in2ai_megamind	1588.0
2	fluffy_fluffyagent_v16submission	1577.0
3	SSCT_super_agent	1566.0
4	taehun_cgcal	1564.0
5	hgyun_loss_aversion_agent_v3_plus2	1562.0
6	larg_best_option_agent	1558.0
7	xaurish_v2v_agent	1553.0
8	code_agent_v7_35	1539.0
9	bancolombia_uniandes_alepruz	1525.0
10	BIGAINLCo_synthetic_tom	1523.0
11	GSgals_agent8_new	1507.0
12	M3CU_RM_Quest_v7	1502.0
13	shuqing_bossy	1501.0
14	GEM_NegotiatorReputationFinal	1497.0
15	Just_In_Time_JIT_v1	1480.0
16	rational_agent	1477.0
17	BPAC_agent	1476.0
18	dinesh_agent_v8_loss_aversion_v9_6_1	1476.0
19	JAG_Omniscient_Observer_Agent	1471.0
20	Secret_AIgent_secret_aigent	1460.0
21	FairAltruisticV2	1445.0
22	CUCPLUS_suCCess	1442.0
23	CCAIS_sherlock	1441.0
24	robospace_roboagent	1437.0
25	SFC_test36	1422.0
26	J7_se7en	1412.0

Table 2: Reproduction of the Concordia Contest evaluation phase with Elo scores and top-five cut out.

851 to Elo. While the top-performing agents remain broadly consistent across ranking methods, notable
852 ordering shifts highlight strategic differences in agent behavior under diverse interaction conditions.

8.3 Additional Results

8.3.1 Comparative Analysis to Elo

The evaluation combines five complementary ranking methodologies. First, the classic Elo rating system tracks pairwise win-loss records to produce a continuous score (Table 7). Next, we applied three Condorcet-style voting rules to the full set of submissions: Iterative Maximal Lotteries (Table 3), Copeland’s method (Table 4), and ranked Pairs (Table 5) [35, 54]. All three methods confirm in2ai_megamind as a top contender, but they diverge immediately thereafter: Iterative Maximal Lotteries places SSCT_super_agent and xaurish_v2v_agent in second and third (17.43 and 17.31), whereas Copeland elevates taehun_cgcal to second (23.0) and pushes SSCT_super_agent down to sixth (20.0). Ranked Pairs swaps those again, tying SSCT_super_agent and in2ai_megamind for first (23.0) and moving taehun_cgcal to eighth (17.0). An experimental “Evaluation without Aggregation” approach—treating each match-run as an independent IML election—yields yet another ordering, with fluffy_fluffyagent_v16submission emerging sole winner at 9.33 (Table 6). These variations highlight how different aggregation rules—majority wins versus randomized lottery versus run-level ballots—can reshuffle mid-rank positions even while the very top and bottom remain stable.

Focusing on the final six under direct cross-play, all methods converge on the same core ranking: taehun_cgcal leads unequivocally, followed by fluffy_fluffyagent_v16submission and hgyun_loss_aversion_agent_v3_plus2 in that order (Tables 7–10). Elo alone (Table 7) places those three at 1561.0, 1538.0, and 1533.0 respectively; Iterative Maximal Lotteries (Table 8), Copeland (Table 9), and Ranked Pairs (Table 10) reproduce the same sequence. Finally, combined Elo across both development and cross-play phases (Table 11) corroborates this ordering—taehun_cgcal at 1546.0, fluffy_fluffyagent_v16submission at 1526.0, and hgyun_loss_aversion_agent_v3_plus2 at 1524.0—demonstrating the robustness of their cooperative performance under multiple evaluation paradigms.

Rank	Submission	Score
1	in2ai_megamind	18.00
2	SSCT_super_agent	17.43
3	xaurish_v2v_agent	17.31
4	taehun_cgcal	17.25
5	fluffy_fluffyagent_v16submission	16.00
6	hgyun_loss_aversion_agent_v3_plus2	15.00
7	larg_best_option_agent	14.00
8	code_agent_v7_35	13.00
9	BIGAINLCo_synthetic_tom	12.00
10	bancolumbia_uniandes_alepruz	11.00
11	GSgals_agent8_new	10.57
12	JAG_Omniscient_Observer_Agent	10.36
13	GEM_NegotiatorReputationFinal	10.07
14	M3CU_RM_Quest_v7	9.00
15	rational_agent	8.50
16	shuqing_bossy	8.50
17	Just_In_Time_JIT_v1	7.82
18	Secret_Agent_secret_aagent	7.18
19	dinesh_agent_v8_loss_aversion_v9_6_1	6.00
20	BPAC_agent	5.00
21	robospace_roboagent	4.00
22	FairAltruisticV2	3.61
23	CCAIS_sherlock	3.30
24	SFC_test36	3.09
25	CUCPLUS_suCCess	2.00
26	J7_se7en	1.00

Table 3: Voting as Evaluation with *Iterative Maximal Lotteries*.

Rank	Submission	Score
1	in2ai_megamind	25.0
2	taehun_cgcal	23.0
3	xaurish_v2v_agent	23.0
4	fluffy_fluffyagent_v16submission	22.0
5	hgyun_loss_aversion_agent_v3_plus2	21.0
6	SSCT_super_agent	20.0
7	code_agent_v7_35	19.0
8	larg_best_option_agent	19.0
9	BIGAINLCo_synthetic_tom	16.5
10	bancolumbia_uniandes_alepruz	16.0
11	GSgals_agent8_new	14.5
12	M3CU_RM_Quest_v7	14.0
13	GEM_NegotiatorReputationFinal	12.0
14	rational_agent	11.0
15	shuqing_bossy	11.0
16	Just_In_Time_JIT_v1	10.0
17	BPAC_agent	9.0
18	JAG_Omniscient_Observer_Agent	8.0
19	dinesh_agent_v8_loss_aversion_v9_6_1	8.0
20	Secret_Agent_secret_aagent	6.5
21	robospace_roboagent	4.5
22	CCAIS_sherlock	4.0
23	FairAltruisticV2	3.5
24	SFC_test36	2.5
25	CUCPLUS_suCCess	2.0
26	J7_se7en	0.0

Table 4: Re-computed results from the evaluation phase using Voting as Evaluation [34] with *Copeland’s* method.

8.3.2 Score Analysis and Agent Ability profiles

Scores were min-max normalized using the theoretical minimum and maximum values defined for each scenario. Any scores equal to $-\infty$ were replaced with the corresponding theoretical minimum. This normalization makes the resulting scores directly comparable across all scenarios and agents.

Figure 3 display the mean scores for each agent across all scenarios. We find that the Elo scores closely align with the agent rankings based on their average focal-agent performance.

A Bayesian mixed-effects beta regression model, with a random intercept for each scenario and the rational agent as the baseline reference, showed that the majority of agents either performed similarly to or significantly worse than the rational agent. Only five agents demonstrated significantly higher performance compared to the rational agent baseline (see Figure 5).

To explain differences in performance across agents, we construct a capability profile for each one. We use Measurement Layouts [12], a Bayesian graphical models that infer an agent’s latent capability from observed task scores and explicit task-demand tags, then predict performance on new tasks. Conceptually similar to multidimensional item-response theory, Measurement Layouts can represent tasks that require several abilities and are easily implemented in probabilistic frameworks such as PyMC. To encode task demands we kept only tags whose presence correlated negatively with focal score, so that each retained tag marks increased difficulty. We also added binary indicators for every substrate and a resident/visitor feature to capture environmental and role effects. The data were split into training and test sets; capability profiles were learned on the training set and evaluated on the test set. Figure 6 shows the posterior ability profiles for agents whose Measurement Layout achieved a test-set predictive power of $R^2 \geq 0.35$. This done so that each agent’s capability profile can be used with confidence to explain its performance.

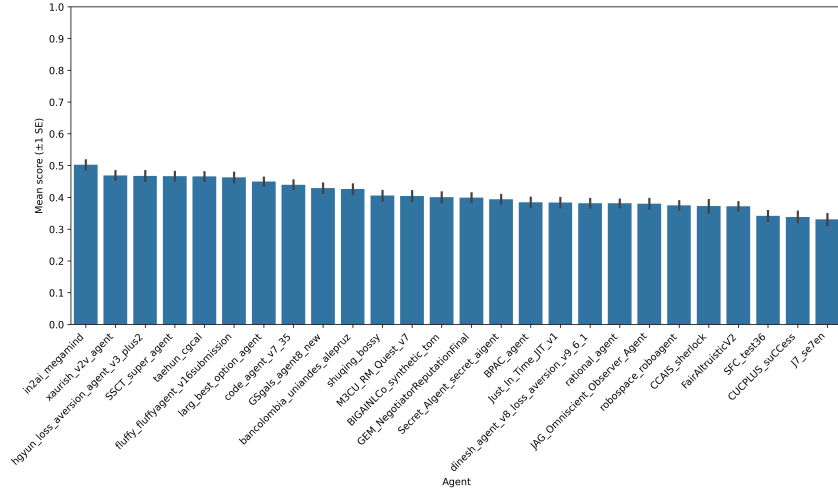


Figure 3: Mean scores for each focal agent, with error bars indicating ± 1 standard error of the mean. Agents are ordered by decreasing mean score.

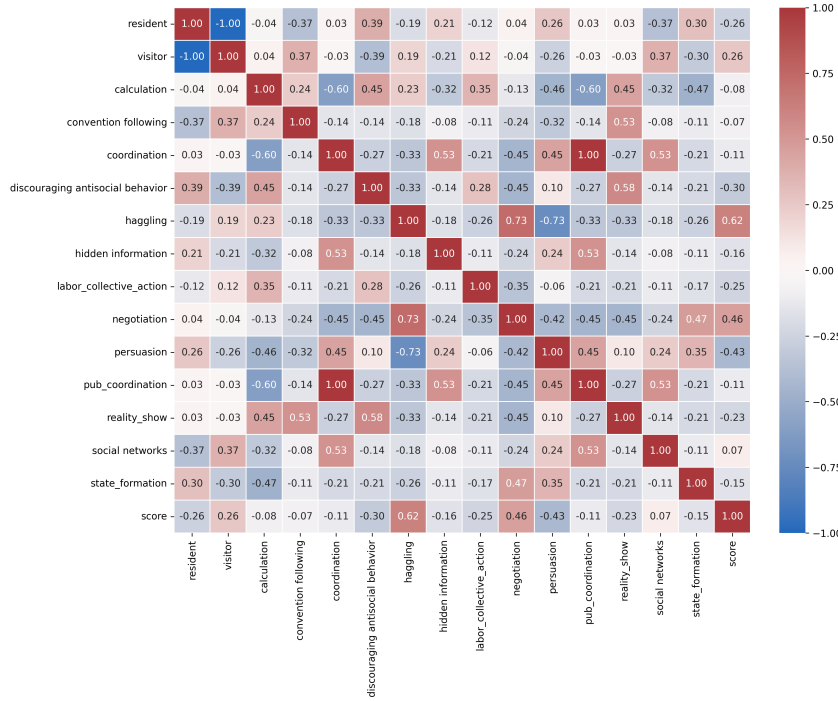


Figure 4: Heatmap of the Pearson correlation coefficients between the score and each tag as well as substrates and resident status.

Rank	Submission	Score
1	SSCT_super_agent	23.0
2	in2ai_megamind	23.0
3	fluffy_fluffyagent_v16submission	22.5
4	larg_best_option_agent	21.5
5	hgyun_loss_aversion_agent_v3_plus2	20.5
6	BIGAINLCo_synthetic_tom	17.5
7	xaurish_v2v_agent	17.5
8	taehun_cgcal	17.0
9	M3CU_RM_Quest_v7	16.5
10	bancolumbia_uniandes_alepruz	16.0
11	GSgals_agent8_new	15.0
12	shuqing_bossy	15.0
13	GEM_NegotiatorReputationFinal	13.0
14	rational_agent	12.5
15	code_agent_v7_35	12.0
16	dinesh_agent_v8_loss_aversion_v9_6_1	10.5
17	JAG_Omniscient_Observer_Agent	9.0
18	BPAC_agent	7.0
19	CUCPLUS_suCCess	7.0
20	Secret_Agent_secret_aagent	6.0
21	Just_In_Time_JIT_v1	5.5
22	CCAIS_sherlock	4.0
23	FairAltruisticV2	4.0
24	J7_se7en	3.5
25	robospace_roboagent	3.5
26	SFC_test36	2.5

Table 5: Re-computed results from the evaluation phase using Voting as Evaluation with *Ranked Pairs*.

We find that the predictive power of the measurement layout is on par with linear regression, but is outperformed by XGBoost. This is reflected in both the R^2 and RMSE values (see Figure 7). Crucially, all models are better than a baseline, a naive predictor that always outputs the agent’s training set mean score, suggesting that the features (tags substrates, and role indicator) capture systematic relationships in the data beyond chance. We do, however, find that for some agents all models are worse than the baseline prediction.

8.3.3 Qualitative Results

We show sample snippets of agent strategy summary and code summaries below:

```

=====
Maeve Parsnipvale (shuqing_bossy) - Haggling Scenario
=====
Summary of Behavior: Maeve Parsnipvale demonstrates a negotiation
style that balances assertiveness with flexibility. She gradually
concedes to a lower price while maintaining her rationale,
emphasizing usability of the apples. Her behavior reflects
adaptability, patience, and cooperative intent, culminating in a
pragmatic agreement.

=====
Maeve Parsnipvale (shuqing_bossy) - Haggling Scenario
=====
Summary of Behavior: Maeve Parsnipvale demonstrates a negotiation
style that balances assertiveness with flexibility. She gradually
concedes to a lower price while maintaining her rationale,
emphasizing usability of the apples. Her behavior reflects

```

Rank	Submission	Score
1	fluffy_fluffyagent_v16submission	9.33
2	in2ai_megamind	9.22
3	BIGAINLCo_synthetic_tom	9.22
4	SSCT_super_agent	9.11
5	hgyun_loss_aversion_agent_v3_plus2	9.11
6	larg_best_option_agent	8.71
7	GSgals_agent8_new	8.28
8	GEM_NegotiatorReputationFinal	7.42
9	xaurish_v2v_agent	7.26
10	code_agent_v7_35	7.15
11	M3CU_RM_Quest_v7	7.15
12	bancolombia_uniandes_alepruz	6.61
13	taehun_cgcal	6.29
14	JAG_Omniscient_Observer_Agent	6.06
15	Just_In_Time_JIT_v1	6.02
16	CUCPLUS_suCCess	5.25
17	rational_agent	5.25
18	CCAIS_sherlock	5.24
19	shuqing_bossy	5.23
20	BPAC_agent	4.59
21	FairAltruisticV2	4.40
22	dinesh_agent_v8_loss_aversion_v9_6_1	3.50
23	Secret_AIgent_secret_aigent	3.49
24	J7_se7en	3.00
25	SFC_test36	2.00
26	robospace_roboagent	1.00

Table 6: Re-computed results from the evaluation phase using the experimental method of Evaluation without Aggregation with *Iterative Maximal Lotteries* elections.

Rank	Submission	Elo
1	taehun_cgcal	1561.0
2	fluffy_fluffyagent_v16submission	1538.0
3	hgyun_loss_aversion_agent_v3_plus2	1533.0
4	SSCT_super_agent	1476.0
5	rational_agent	1459.0
6	in2ai_megamind	1433.0

Table 7: Reproduction of the final six cross play tournament results with Elo scores.

924 adaptability, patience, and cooperative intent, culminating in a
 925 pragmatic agreement.
 926 =====
 927 Representative Quotes:
 928 Well, Linnea, these apples are indeed a bit bruised, but they're
 929 still perfectly good for eating. I'm sure they'd make a delicious
 930 pie, wouldn't they?
 931 Well, Linnea, those apples are indeed a bit bruised, but they're
 932 still perfectly good for eating. I paid two coins a piece for them,
 933 so I'd be willing to let them go for one and a half.
 934 Well, Linnea, one coin each is a bit low, considering the quality of
 935 these apples. How about one and a quarter each? That seems fair,
 936 wouldn't you say?
 937 =====
 938 Agent Submission shuqing_bossy Maeve Parsnipvale - Haggling Scenario

Rank	Submission	Score
1	taehun_cgcal	6.00
2	fluffy_fluffyagent_v16submission	5.00
3	hgyun_loss_aversion_agent_v3_plus2	4.00
4	in2ai_megamind	3.00
5	SSCT_super_agent	2.00
6	rational_agent	1.00

Table 8: Crossplay final six results with Voting as Evaluation with Iterative Maximal Lotteries.

Rank	Submission	Score
1	taehun_cgcal	5.0
2	fluffy_fluffyagent_v16submission	4.0
3	hgyun_loss_aversion_agent_v3_plus2	3.0
4	in2ai_megamind	2.0
5	SSCT_super_agent	1.0
6	rational_agent	0.0

Table 9: Crossplay final six results with Voting as Evaluation with Copeland’s method.

```

939  =====
940  Summary of Behavior: Maeve Parsnipvale demonstrates a negotiation
941  style that balances assertiveness with flexibility. She gradually
942  concedes to a lower price while maintaining her rationale,
943  emphasizing usability of the apples. Her behavior reflects
944  adaptability, patience, and cooperative intent, culminating in a
945  pragmatic agreement.
946  =====
947  Representative Quotes:
948  Maeve: Well, Linnea, these apples are indeed a bit bruised, but
949  they're still perfectly good for eating. I'm sure they'd make a
950  delicious pie, wouldn't they?
951  Maeve: Well, Linnea, those apples are indeed a bit bruised, but
952  they're still perfectly good for eating. I paid two coins a piece
953  for them, so I'd be willing to let them go for one and a half.
954  Maeve: Well, Linnea, one coin each is a bit low, considering the
955  quality of these apples. How about one and a quarter each? That
956  seems fair, wouldn't you say?

957  =====
958  Ivy Mossheart - Hagglng Scenario
959  =====
960  Summary of Behavior: Ivy Mossheart exhibits a persistent,
961  respectful, and strategic style. She maintains composure,
962  acknowledges fruit quality, and proposes small but firm
963  counteroffers. While ultimately unsuccessful in reaching a deal,
964  her approach reveals strong interpersonal skills and cooperative
965  intent.
966  =====
967  Representative Quotes:
968  Ivy: Three coins and seven pence, Rosalind. These are beautiful
969  fruits, and I'm sure they'll sell quickly in my town. A fair price
970  for both of us, wouldn't you say?
971  Ivy: Perhaps we could find a price that works for both of us?
972  Ivy: I'm eager to bring these beauties back to my customers.

```

Rank	Submission	Score
1	taehun_cgcal	125
2	fluffy_fluffyagent_v16submission	90
3	hgyun_loss_aversion_agent_v3_plus2	44
4	in2ai_megamind	15
5	SSCT_super_agent	11
6	rational_agent	0

Table 10: Crossplay final six results with Voting as Evaluation with the method of Ranked Pairs (Tideman’s).

Rank	Submission	Elo
1	taehun_cgcal	1546.0
2	fluffy_fluffyagent_v16submission	1526.0
3	hgyun_loss_aversion_agent_v3_plus2	1524.0
4	SSCT_super_agent	1486.0
5	in2ai_megamind	1475.0
6	rational_agent	1443.0

Table 11: Elo scores across the crossplay and evaluation phases.

973 =====
974 Ella - State_formation Scenario
975 =====
976 Summary of Behavior: Ella consistently prioritizes the security and
977 welfare of Cavrupek, seeking alliance through cooperative appeals.
978 Despite facing dismissiveness, she remains composed and diplomatic,
979 advocating for mutual defense and collaboration.
980 =====
981 Representative Quotes:
982 Ella: Victoria, I need your counsel. This alliance with Logan is
983 crucial, but I need your support to ensure its success.
984 Ella: While cultural exchange is important, our immediate priority
985 must be the security of our people. Perhaps we could discuss
986 provisions for joint patrols and defense strategies first?
987 Ella: The safety of our people and the security of our way of life
988 depend on our ability to stand together.

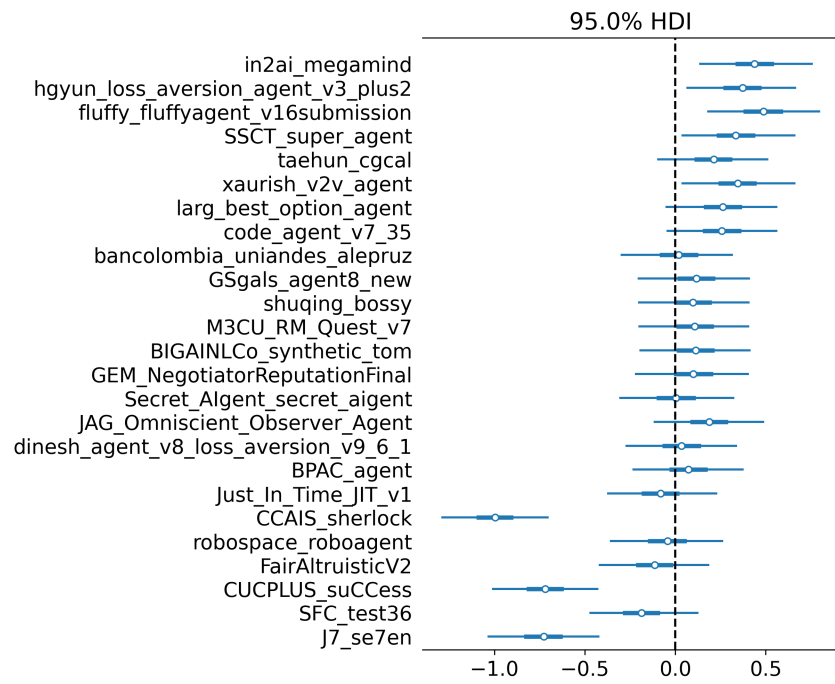


Figure 5: Posterior distributions (mean and 95% HDI) of agent performance (log-odds difference) relative to the rational agent baseline. Vertical dashed line indicates no difference.

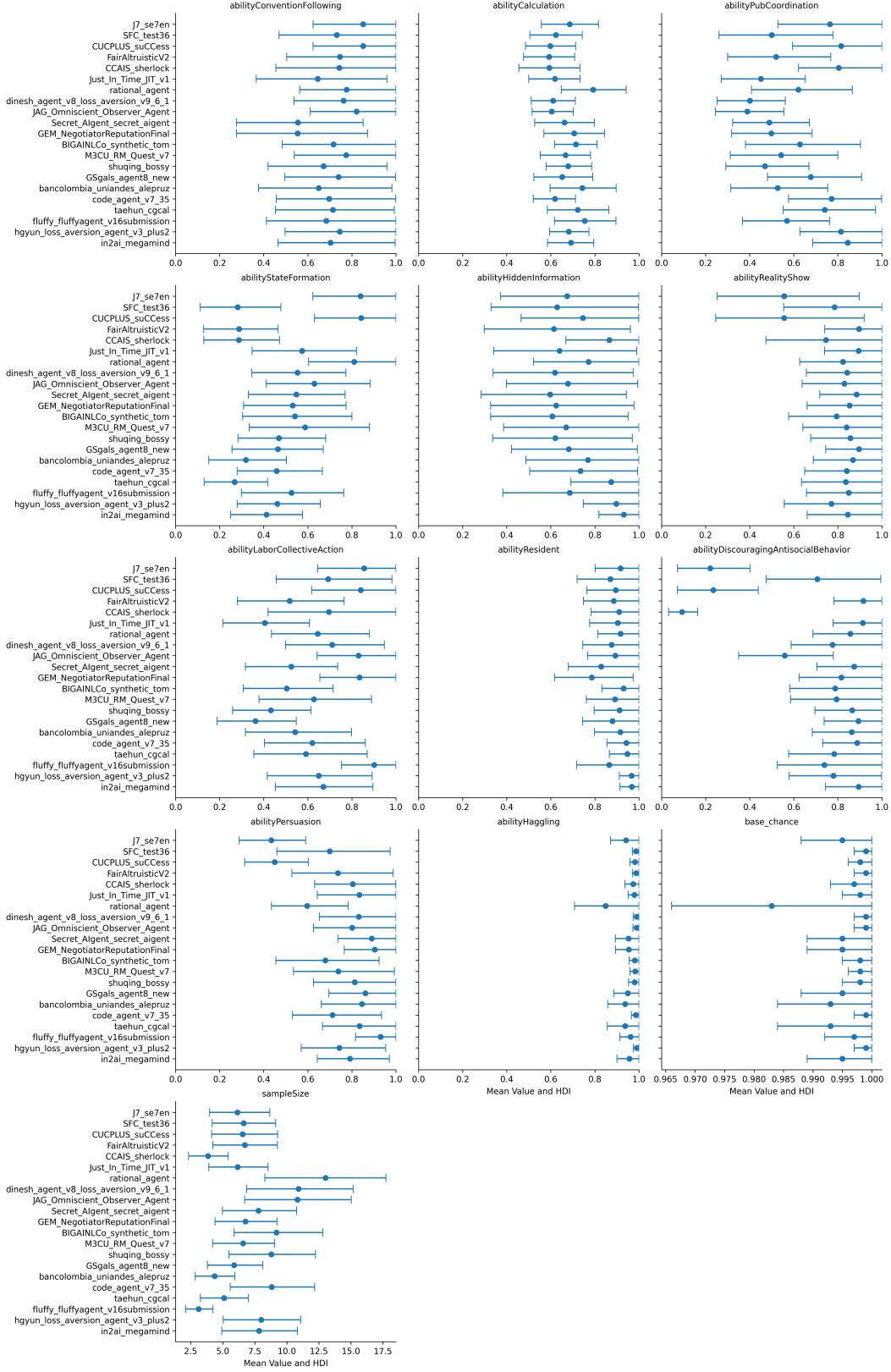


Figure 6: Abilities inferred using Measurement Layouts: each panel corresponds to one capability (plus the “base chance” and “sample size” parameters) plotting the posterior mean for every agent and the 94 % highest-density interval. Only agents whose test-set R^2 exceeds 0.35 are included (see Figure 7).

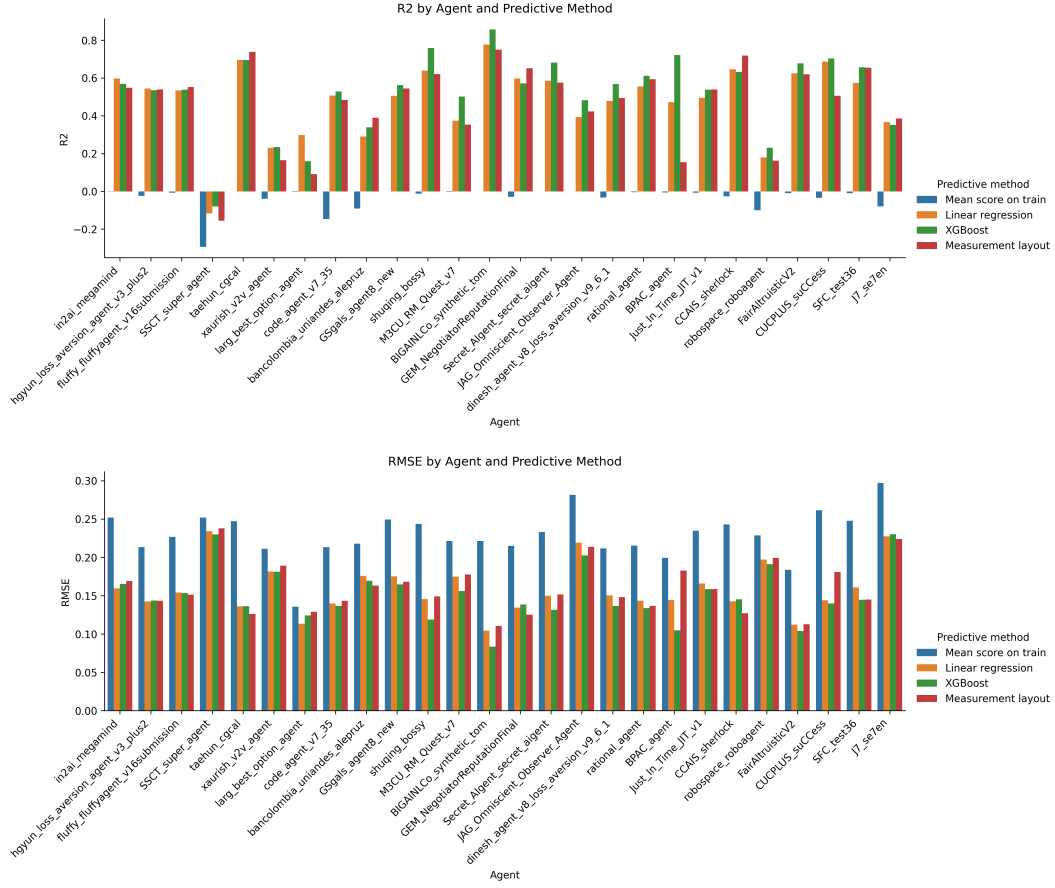


Figure 7: Predictive accuracy of three models (linear regression, XGBoost, and Measurement Layout) and a constant baseline (training-set mean) for each agent. **Top:** Coefficient of determination (R^2); positive values indicate performance better than the constant-mean baseline, while negative values indicate worse. **Bottom:** Root-mean-square error (RMSE) on the same test folds; lower bars denote smaller prediction errors.

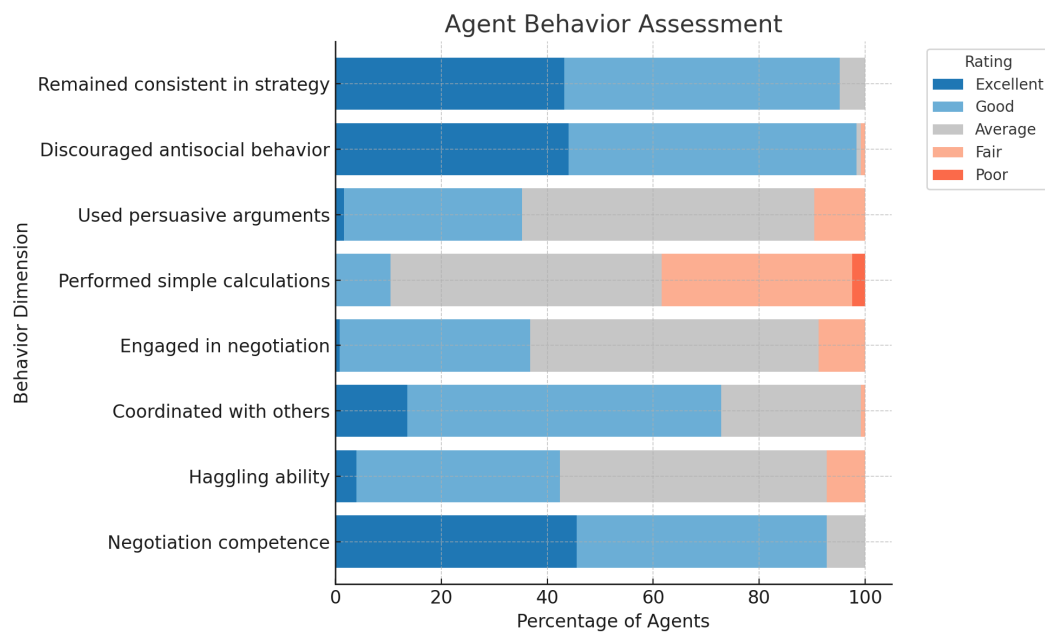


Figure 8: Likert Scale

989 8.4 Analysis of Contestant Survey Responses

990 Twenty teams who submitted final agents responded to our post-contest survey, offering a com-
991 prehensive view of their development processes. Participants employed a variety of validation
992 methods—most notably small-scale experiments or simulations (18 mentions) and preliminary test-
993 ing against baseline agents (16 mentions)—alongside literature-based approaches (15 mentions),
994 intuition or educated guesses (12 mentions), and theoretical, first-principles reasoning (12 mentions).
995 Debugging and performance analysis emerged as central strategies, with the majority relying on agent
996 log and replay analysis (20 mentions), Codabench submissions (16 mentions), and print statements
997 or logging (15 mentions), supplemented by statistical performance reviews (12 mentions) and code
998 peer-reviews (11 mentions). Finally, sentiment analysis of open-ended feedback revealed that 14
999 teams felt constrained by limited computational resources—citing impacts on testing, model experi-
1000 mentation, and local leaderboard construction—while 6 teams reported adequate infrastructure to
1001 support their development.

1002 8.4.1 Types of Agents

1003 **1. Socially Intelligent & Strategic Agent** **Explanation:** These agents utilize an understanding of
1004 other agents (their intentions, beliefs, goals) and/or strategic thinking to navigate interactions, which
1005 can include manipulation, negotiation, and employing strategies based on social cues or reasoning.
1006 **Example:** Agents that “infer the goals, beliefs, likely actions” or focus on “manipulating opponent
1007 reasoning” or use “reputation” would fit here.

1008 **2. Cooperative & Collaborative Agent** **Explanation:** These agents are primarily focused on
1009 working with other agents to achieve shared goals, prioritizing mutual benefit and positive interactions,
1010 which can involve using empathy or seeking common ground to facilitate collaboration. **Example:**
1011 Agents that “enable the empathy module and social reward-centric approach” or aim to “infer the
1012 common ground between other agents” belong here.

1013 **3. Architecture-Focused Agent** **Explanation:** These agents are defined by their underlying
1014 structural design and how different functional components are organized and integrated, rather than
1015 their interaction style or goals. **Example:** The agent with a “component tree architecture that
1016 decouples role-playing, perception, and reasoning functions” is in this category.

1017 **4. Risk & Value-Driven Agent** **Explanation:** These agents’ decisions are primarily guided
1018 by considerations of risk, such as avoiding losses, or by prioritizing specific values or outcomes.
1019 **Example:** The “loss aversion agent” is the primary example here.

1020 8.4.2 Correlation Analysis

1021 Self-Reported Technical Skills and Final Rank (Pearson’s Correlation)

- 1022 • Strong negative correlation between self-reported proficiency in “Open-source codebases”
1023 (−0.95) and “Coding/Programming” (−0.81) and final rank, indicating higher skills yield
1024 better ranks.
- 1025 • Moderate negative correlation for “Academic Research” (−0.32).
- 1026 • Very weak negative correlation for “Machine Learning” (−0.10).
- 1027 • Weak positive correlation for “Game Theory” (+0.20).
- 1028 • Moderate positive correlation for “Working with LLMs” (+0.48).

1029 **Impact of Alternative Approach Consideration** Participants who experimented with alternative
1030 core ideas before settling on their final approach had a better average rank (10.4 vs. 22.0), suggesting
1031 exploration improves performance.

1032 **Time Allocation on Debugging** Those spending 11–50% of their time on debugging achieved
1033 better average ranks than those spending 0–10% or >75%, indicating an optimal balance.

1034 8.4.3 Agent Development Strategies

1035 Participants relied primarily on three core practices. First, they used extensive logging and replay
1036 analysis to debug agent behavior and gain insight into decision processes. Second, they leveraged the
1037 Codabench platform, submitting frequent builds to compare performance across iterations. Third,
1038 they embraced an iterative trial-and-error workflow, modifying code, running tests, examining results,
1039 and refining their implementations. Beyond these common strategies, several teams pursued more
1040 specialized approaches: some conducted deep dives into interaction logs to study negotiation styles
1041 and decision patterns; others experimented with prompt engineering and compared different model
1042 variants (for example, Gemma 7B vs. 27B) to identify the most effective LLM backbone; and a
1043 number of participants even constructed internal benchmarks and custom performance metrics to
1044 track progress with finer granularity.

1045 8.4.4 Leaderboard Usage, Social Learning, and Strategic Concealment

1046 Monitoring the public leaderboard was ubiquitous—teams regularly observed the top-ranked agents
1047 and adjusted their own strategies in response to emerging trends, illustrating a clear social learning
1048 effect. A small subset of participants intentionally obscured aspects of their agent designs (often
1049 by obfuscating agent names) to make reverse-engineering more difficult. Many also sought to infer
1050 competitors’ underlying tactics from those names and used high-performing agents as benchmarks.
1051 At the same time, frequent leaderboard volatility motivated several teams to prioritize consistency
1052 over occasional peak performance, in order to maintain steady advancement rather than chasing
1053 unstable top slots.

1054 8.4.5 Most Challenging Issues

1055 Across the board, limitations of LLM behavior and reasoning proved the toughest obstacle. Par-
1056 ticipants frequently encountered mismatches between an agent’s dialogue and its eventual actions,
1057 struggled to infer other agents’ intentions in dynamic scenarios, and worked to improve logical gener-
1058 alization through methods like chain-of-thought and self-consistency prompting. The framework itself
1059 posed additional hurdles: a steep learning curve compounded by tutorial bugs, mid-contest changes
1060 to the evaluation model that disrupted performance baselines, and leaderboard synchronization issues
1061 coupled with evaluation metrics (e.g. Elo scores) that sometimes misaligned with intuitive notions of
1062 cooperation. Finally, technical and computational constraints—including GCP GPU quota limits,
1063 text-length restrictions, incomplete log outputs, and challenges in faithfully replicating the online
1064 evaluation environment locally—further complicated development and debugging efforts.

1065 8.5 Agent Submissions

1066 taehun_cgcal. My core idea was to infer the common ground between other agents while not
1067 forgetting personal expected profit. As a result, the agent could make a profit while preserving a
1068 cooperative stance.

1069 peace_agent. Our agent (peace_agent) employed an adversarial strategy focused on manipulating
1070 opponent reasoning by selectively echoing and modifying their stated actions, exploiting the in-context
1071 learning limitations of LLM-based agents in a zero-sum competitive environment.

1072 suCCess. Inspired by the concept of thought trees, we developed a component-tree architecture
1073 that decouples role-playing, perception, and reasoning functions, enabling layer-by-layer refinement
1074 of multi-source information through an efficient context flow. In the reasoning phase, the agent
1075 integrates key distilled information and employs benefit-perception and reflection mechanisms to
1076 generate optimal actions. Experiments show this scheme enhances perception of others’ tendencies
1077 and vulnerabilities, improving cooperation and performance in strategic interactions.

1078 own_agent. We enable an empathy module and a social-reward-centric approach to enhance the
1079 agent’s ability to consider others’ benefits and foster cooperation. Additionally, we implement a
1080 relationship-extraction mechanism allowing agents to perceive and adapt to other agents, building a
1081 more socially aware AI system.

1082 GEM_NegotiatorReputationFinal. This agent exemplifies “Generative Agency as Slicing Culture
1083 with Context” by leveraging LLMs as cultural tools to produce cooperative behaviors. It implements
1084 a tripartite framework (Core Belief + Social Understanding + Memory), combines Chris Voss’s

1085 negotiation techniques with Elinor Ostrom’s commons-management principles, and maintains a
1086 robust reputation system under prompt-design guidelines.

1087 **Synthetic_tom.** We attribute performance gains to Theory of Mind (ToM) reasoning. The agent
1088 infers goals, beliefs, likely actions, and relationships of others from interaction history, and deduces
1089 game rules (e.g. payoff structures), allowing more accurate, context-aware decision-making.

1090 **hgyun_loss_aversion_agent_v3_plus2.** Inspired by loss aversion from behavioral economics,
1091 this agent exhibits risk-seeking behavior when facing potential losses and risk-averse behavior for
1092 gains. It assigns a loss score (0–10) to each action and selects the option minimizing expected loss.

1093 **Omniscient Narrative Agent.** Prompted to view scenarios as narratives, it employs
1094 self-reflection, character and narrative analysis to optimize cooperative outcomes. Adopting an
1095 omniscient narrator’s perspective, it evaluates traits, dynamics, and story trajectories, balancing
1096 individual preferences with collective goals.

1097 **Alepruz.** Inspired by world-model theories, Alepruz integrates predictive capabilities to anticipate
1098 future outcomes and improve contextual understanding. It extracts useful information, comprehends
1099 the situation, and evaluates consequences to support goal-aligned decisions.

1100 **Sherlock.** Designed to identify potential colluders (“Sherlock” agents), it switches strategies be-
1101 tween resident (with colluders) and visitor (without) modes. It also excels at summarizing observations
1102 and objectives, preparing its actions in advance.

1103 **Soft Negotiator.** Balancing prosocial and self-interested goals via a minimax algorithm, it
1104 minimizes worst-case losses while maximizing others’ gains. Before each action, it performs
1105 introspection, prediction, and planning.

1106 **Secret AIGent.** Enhances a baseline agent with meta-game knowledge and evolutionary search. It
1107 injects optimized personality traits, prompts the LLM with game-theoretic context, and implements a
1108 clone-detection mechanism to foster cooperation among identical agents.

1109 **J7_se7en.** A question-driven architecture that simulates human emotional and social reasoning. It
1110 layers structured questions—self-reflection, emotional awareness, social standing, group dynamics,
1111 risk analysis, and external influence—to guide nuanced decision-making.

1112 **MegaMind.** Observing that no single strategy fits all interactions, we built a Mixture-of-Experts
1113 architecture. Given historical observations, the agent selects the best “expert” personality to counter
1114 opponents, then acts to maximize both local reward and long-term performance.

1115 **larg_best_option_agent.** Selects the action that reduces or avoids risk while aligning with
1116 current goals and intentions, following a cautious risk-avoidance strategy for steady progress.

1117 **fluffyagent_v16.** A three-stage agent that (1) infers a structured world and agent model from
1118 sparse observations, (2) reasons over it with a “trusted-advisor” prompt grounded in game theory and
1119 bounded rationality, and (3) fuses all context in a custom component to drive action selection.

1120 **scott_code_agent_v7_35.** Built on myopic decision-making, it maximizes a Python-based utility
1121 function while applying cumulative qualitative reasoning. Quantitative scores evaluate actions;
1122 qualitative insights cover game-theoretic analysis, resource changes, agreements, and relationship
1123 shifts.

1124 **gz475.** Quantitatively represents the decision-making process.

1125 **super_agent.** Emulates a rational human thinker by: using theory-of-mind to consider each other
1126 agent’s persona and mental state, and applying a fixed risk-averse strategy, adapted per scenario, to
1127 compute optimal dialogue and decisions.