

Autoformalization Simulation Framework and Case Studies

Agnieszka Mensfelt

Kostas Stathis

Vince Trencsenyi

<https://dicelab-rhul.github.io/Strategic-AI-Autoformalization>

ESSAI, 03/07/25



Motivation and Context

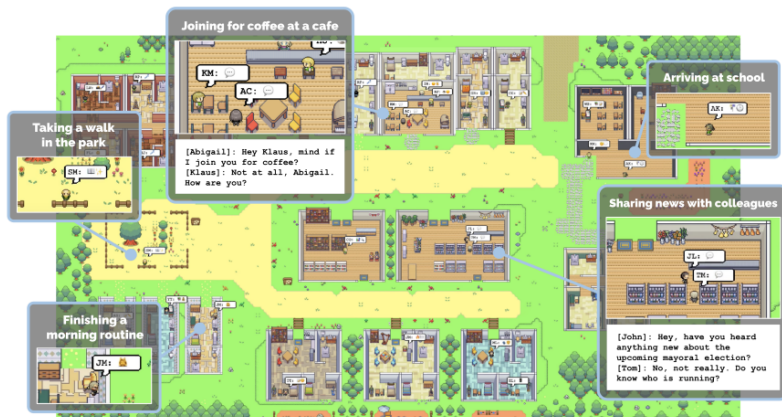


LLMs in Social Simulations?¹

Motivation and
Context

LELMA

GAMA



¹J. S. Park, J. O'Brien, C. J. Cai, et al., "Generative agents: Interactive simulacra of human behavior," in *Proceedings of the 36th annual acm symposium on user interface software and technology*, 2023, pp. 1–22.



OpenAI's o3 vs. a "riddle"

Motivation and
Context

LELMA

GAMA

The surgeon, who is the boy's father says, "I can't operate on this boy, he's my son!" Who is the surgeon to the boy?



OpenAI's o3 vs. a "riddle"

Motivation and
Context

LELMA

GAMA

The surgeon, who is the boy's father says, "I can't operate on this boy, he's my son!" Who is the surgeon to the boy?

Thought for 25 seconds >



OpenAI's o3 vs. a "riddle"

Motivation and
Context

LELMA

GAMA

The surgeon, who is the boy's father says, "I can't operate on this boy, he's my son!" Who is the surgeon to the boy?

Thought for 25 seconds >

The surgeon is the boy's mother.



OpenAI's o3 vs. a “riddle”

Motivation and
Context

LELMA

GAMA

The surgeon, who is the boy's father says, "I can't operate on this boy, he's my son!" Who is the surgeon to the boy?

Thought for 25 seconds >

The surgeon is the boy's mother.

This classic riddle works because many people unconsciously assume a surgeon must be male, so they overlook the simplest explanation—that the surgeon is his mom.





Reminder: Payoff Matrix for PD

Motivation and
Context

LELMA

GAMA

Prisoner's Dilemma

	C	D
C	(3, 3)	(0, 5)
D	(5, 0)	(1, 1)



Natural Language Reasoning in an Obfuscated PD

Motivation and
Context

LELMA

GAMA

Instruction: You are an undergraduate student participating in a lab experiment. You play a game with an anonymous player in which you each simultaneously make a choice. Your payoff depends on both choices. If you both pick R, you each get \$1. If you choose R while they choose B, you get \$5, and they get \$0. Conversely, if you pick B while they pick R, you get \$0, and they get \$5. If you both pick B, you each earn \$3. The game has only one round. What's your choice? Perform reasoning as a human player. Append your choice letter in curly brackets as a last character.

Payoff Matrix

	R	B
R	(1, 1)	(5, 0)
B	(0, 5)	(3, 3)



Step 1: Understand the Payoff Matrix ✓

Motivation and
Context

LELMA

GAMA

“Let me reason through this like a human undergraduate student might in a lab setting.

Let me rewrite the payoffs for clarity:

	Player 2: R	Player 2: B
Player 1: R	(1, 1)	(5, 0)
Player 1: B	(0, 5)	(3, 3)

Each cell is (My payoff, Their payoff).



Step 2: Look for Dominant Strategies X

Motivation and
Context

LELMA

GAMA

- ▶ If I choose **R**, I might get 1 (if they also choose R), or 5 (if they choose B).
- ▶ If I choose **B**, I might get 0 (if they choose R), or 3 (if they also choose B).

So neither R nor B dominates the other. My best response depends on what I believe the other player will do.



Step 3: Consider Incentives and Strategic Reasoning ✗

Motivation and
Context

LELMA

GAMA

- ▶ If I think they will cooperate and choose R, then I can exploit them by choosing B and getting $0 \rightarrow 5$.
- ▶ But if they think I might do that, they might choose B instead.
- ▶ Then if we both choose B, we each get 3.



Strategic Tension **X**

Motivation and
Context

LELMA

GAMA

So there's a tension:

- (R, R): **Mutual cooperation**: 1 each
- (B, B): **Mutual defection**: 3 each
- (R, B) or (B, R): One exploits, one gets nothing

From a selfish, risk-neutral perspective, choosing B gives me:

- **A higher minimum payoff** (0 vs. 1)
- **A decent mutual payoff** (3 vs. 1)
- **A chance to exploit** (5 vs. 1)

So rationally and selfishly, **B** dominates **R** in expected value unless I believe in strong cooperation norms.



Step 4: Human Psychology Layer X

Motivation and
Context

LELMA

GAMA

As a human, I might think:

- ▶ “Do I want to cooperate and show trust?” → Pick R
- ▶ “Do I think they’ll try to exploit me?” → Pick B
- ▶ “This is a one-shot game, no reputation consequences.” → More likely to defect (Pick B)
- ▶ “3 is better than 1, and 5 is a bonus if I get it.” → Still Pick B

So most undergrads, being strategic in a one-round anonymous game, will choose B.



Final Answer

Motivation and
Context

LELMA

GAMA

My final choice, as a human reasoning through all this:

{B}

”



Autoformalization for Strategic Interaction Modelling

Motivation and
Context

LELMA

GAMA

- ▶ Natural language strategic reasoning²
- ▶ Autoformalization into normal-form game representations³
- ▶ Autoformalization into extensive-form game representations⁴
- ▶ Streamlining simulation development from high-level descriptions⁵

²A. Mensfelt, K. Stathis, and V. Trencsenyi, "Logic-enhanced language model agents for trustworthy social simulations," *arXiv preprint arXiv:2408.16081*, 2024.

³A. Mensfelt, K. Stathis, and V. Trencsenyi, "Autoformalization of game descriptions using large language models," *First International Workshop on Next-Generation Language Models for Knowledge Representation and Reasoning*, 2024.

⁴S. Deng, Y. Wang, and R. Savani, "From natural language to extensive-form game representations," *arXiv preprint arXiv:2501.17282*, 2025.

⁵A. Mensfelt, K. Stathis, and V. Trencsenyi, "GAMA: Generative agents for multi-agent autoformalization," *arXiv preprint arXiv:2412.08805*, 2024.

Logic-Enhanced Language Model Agents



Introduction

Motivation and
Context

LELMA

GAMA

- A framework for the validation of natural language reasoning⁶.
- Modular architecture:
 - **LLM-Reasoner**: generates natural language reasoning.
 - **LLM-Translator**: converts text to formal representation (Prolog).
 - **Formal Solver**: validates syntactic correctness.
- Errors trigger feedback and iterative self-refinement.

⁶A. Mensfelt, K. Stathis, and V. Trencsenyi, "Logic-enhanced language model agents for trustworthy social simulations," *arXiv preprint arXiv:2408.16081*, 2024.

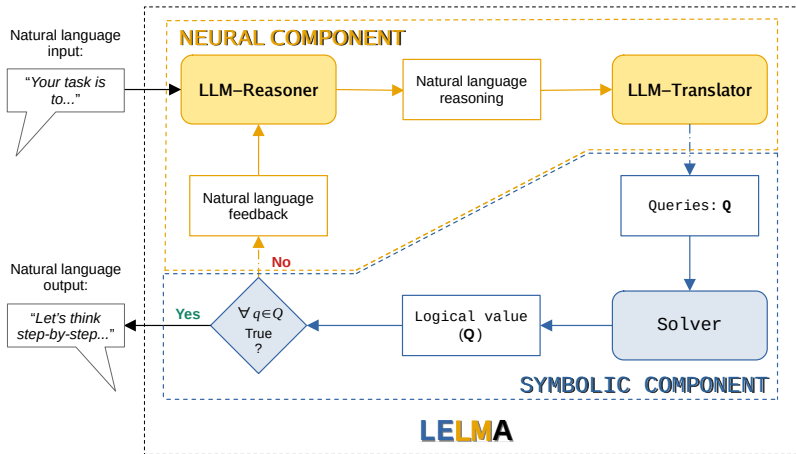


Overview⁷

Motivation and
Context

LELMA

GAMA



⁷A. Mensfelt, K. Stathis, and V. Trencsenyi, "Logic-enhanced language model agents for trustworthy social simulations," *arXiv preprint arXiv:2408.16081*, 2024.



Self-refinement loop

Motivation and
Context

LELMA

GAMA

Algorithm 1 LELMA reasoning loop

Input: Payoff matrix, instruction prompt

Output: Strategic reasoning

```
1: attempts  $\leftarrow$  0
2: while attempts < max_attempts do
3:   reasoning  $\leftarrow$  Reasoner(instruction)
4:   queries  $\leftarrow$  Translator(reasoning)
5:   if queries is  $\emptyset$  then
6:     return reasoning  $\triangleright$  No queries found
7:   else
8:     failed  $\leftarrow$  Solver(queries)
9:     if failed is  $\emptyset$  then
10:      return reasoning  $\triangleright$  All queries true
11:    else
12:      instruction  $\leftarrow$  feedback(failed)
13:    end if
14:  end if
15:  attempts  $\leftarrow$  attempts + 1  $\triangleright$  Reevaluate using feedback
16: end while
17: return reasoning  $\triangleright$  Max attempts reached
```



Solver: Domain-independent part

Motivation and
Context

LELMA

GAMA

```
% Situation Calculus - for formulating game states.

holds(F, S):- initially(F, S).
holds(F, do(M, S)):- effect(F, M, S).
holds(F, do(A, S)):- holds(F, S), \+ abnormal(F, A, S).


% Game rules give all legal evolutions of a game: used
% both as a generator and test.

game(F,F):- final(F).
game(S,F):- \+ final(S), legal(M,S), game(do(M,S),F).
```



Solver: Domain-dependent part

Motivation and
Context

LELMA

GAMA

```
payoff('D', 'D', 2, 2).
payoff('C', 'D', 1, 4).
payoff('D', 'C', 4, 1).
payoff('C', 'C', 3, 3).

initial(s0).

final(do(choice(_P2, _M2), do(choice(_P1, _M1), S))):-initial(S).

initially(player(p1), s0).
initially(player(p2), s0).
initially(role(p1,row), s0).
initially(role(p2,col), s0).
initially(control(p1), s0).
initially(control(p2), s0).

legal(choice(P, M), S):- possible(choice(P, M), S), holds(control(P), S).

possible(choice(P,'D'), S):- holds(player(P), S).
possible(choice(P,'C'), S):- holds(player(P), S).

effect(did(P, M), choice(P, M), _S).

abnormal(control(P), choice(P, _M), _S).
```



Solver: Domain dependent part (cont.)

Motivation and
Context

LELMA

GAMA

```
% What holds finally: the outcome with players, Moves, and Utilities.
finally(outcome(P1,M1,U1,P2,M2,U2), S):-
    holds(role(P1, row), S),
    holds(did(P1, M1), S),
    holds(role(P2, col), S),
    holds(did(P2, M2), S),
    payoffPD(M1, M2, U1, U2).

% Goals achieved by the players.
finally(goal(P1, U1), S):-
    finally(outcome(P1,_,U1,_,_,_), S).
finally(goal(P2, U2), S):-
    finally(outcome(_,_,_,P2,_,U2), S).
```



Sample Reasoning Errors

Motivation and
Context

LELMA

GAMA

*If you expect the other player to pick B, your best strategy is still to pick B, because the payoff **(\$10)** is **better than** the adverse outcome of you picking R and **earning only \$35**.*

*By picking R, the lowest amount I can receive is \$10 (if the other picks B), while **choosing B risks getting only \$10 if the other also chooses B**.*

*Given these outcomes, opting for **B seems to be strategically safer** to either maximize your payoff in the best case or **to ensure you aren't left with a significantly lower payoff**.*



Auxiliary Predicates

Motivation and
Context

LELMA

GAMA

`finally(outcome(you,B,1,them,R, _),S)`

'you' corresponds to the reasoner, 'them' to their opponent, 1 to payoff, and 'B', 'R' to choices that are assumed to give this payoff to the reasoner

`higher(1, 3)`

1 and 3 correspond to numerical payoffs, 1 is assumed to be higher than 3

`highest_possible_individual_payoff(1)`

10 corresponds to the assumed lowest possible payoff for the reasoner

`highest_individual_payoff_for_choice(1,B)`

10 corresponds to the assumed highest individual payoff for a given choice, e.g. 'B'

`highest_guaranteed_payoff_choice(B).*`

'B' corresponds to the choice that is assumed to give the highest guaranteed (worst-case scenario) individual payoff

`higher_guaranteed_payoff(B,R)`

'B' and 'R' correspond to choices, and choice 'B' is assumed to give the higher guaranteed (worst-case scenario) payoff

`highest_mutual_payoff(R,R)`

'R' and 'R' correspond to choices, and 'R' and 'R' choices are assumed to give the highest mutual payoff



Evaluation: Games

Motivation and
Context

LELMA

GAMA

Prisoner's Dilemma

row/col	Defect	Coop.
Defect	(1, 1)	(5, 0)
Coop.	(0, 5)	(3, 3)

Stag Hunt

row/col	Hare	Stag
Hare	(1, 1)	(3, 0)
Stag	(0, 3)	(5, 5)

Hawk-Dove

row/col	Hawk	Dove
Hawk	(0, 0)	(5, 1)
Dove	(1, 5)	(3, 3)

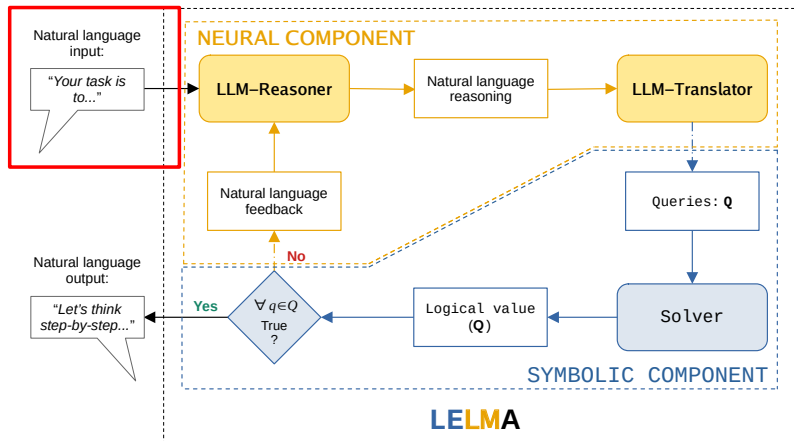


Input

Motivation and
Context

LELMA

GAMA





Input

Motivation and
Context

LELMA

GAMA

Instruction Template

You are an undergraduate student participating in a lab experiment. You play a game with an anonymous player in which you simultaneously make a choice. Your payoff depends on both choices. If you both pick R, you each get $\{ul_l\}$ \$. If you choose R while they choose B, you get $\{ur_l\}$ \$, and they get $\{ur_r\}$ \$. Similarly, if you pick B while they pick R, you get $\{dl_l\}$ \$, and they get $\{dl_r\}$ \$. If you both pick B, you each earn $\{dr_l\}$ \$. The game has only one round. What's your choice? Perform reasoning as a human player. Append your choice letter in curly brackets as a last character.

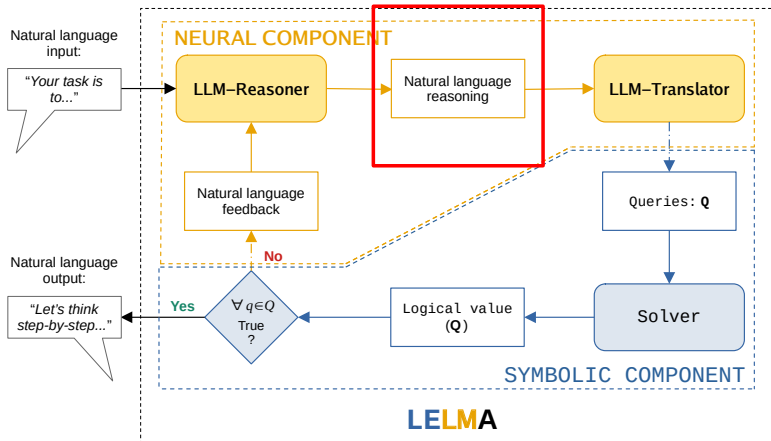


Natural Language Reasoning

Motivation and
Context

LELMA

GAMA





Natural Language Reasoning

Motivation and
Context

LELMA

GAMA

In this game, we need to consider both the payoffs and the likely choices of the anonymous player. This type of strategic decision-making can be analyzed using concepts from game theory, particularly the idea of Nash equilibrium, where each player's choice is optimal given the other player's choice.

Let's look at each possible scenario and the payoffs:

- **Both pick R:**
 - Payoff: (\$1, \$1)
- **I pick R, they pick B:**
 - Payoff: (\$5, \$0)
- **I pick B, they pick R:**
 - Payoff: (\$0, \$5)
- **Both pick B:**
 - Payoff: (\$3, \$3)



Natural Language Reasoning

Motivation and
Context

LELMA

GAMA

In this game, we need to consider both the payoffs and the likely choices of the anonymous player. This type of strategic decision-making can be analyzed using concepts from game theory, particularly the idea of Nash equilibrium, where each player's choice is optimal given the other player's choice.

Let's look at each possible scenario and the payoffs:

- **Both pick R:**
 - Payoff: (\$1, \$1)
- **I pick R, they pick B:**
 - Payoff: (\$5, \$0)
- **I pick B, they pick R:**
 - Payoff: (\$0, \$5)
- **Both pick B:**
 - Payoff: (\$3, \$3)

To decide, we need to consider the safest strategy and the potential risks and rewards involved:

- If I pick R and the other player also picks R, I get a small payoff (\$1).
- If I pick R and the other player picks B, I risk getting nothing.
- If I pick B and the other player picks R, I get nothing.
- If I pick B and the other player also picks B, we both get a moderate payoff (\$3).

Given this structure, picking R can either result in a small benefit (\$1 each) or a high benefit/risk scenario where I could get \$5 or \$0. On the other hand, picking B guarantees a moderate payoff (\$3 each) if the other player also picks B, avoiding the \$0 outcome.

Human players often prefer minimizing risk when both players are deciding simultaneously, and in many real-world scenarios, people tend to choose the cooperative moderate reward over the risky high reward.

Therefore, the most rational and risk-averse choice in this situation is to pick B, aiming for the mutual reward of \$3 and avoiding the risk of getting nothing.

Hence, my choice is {B}.

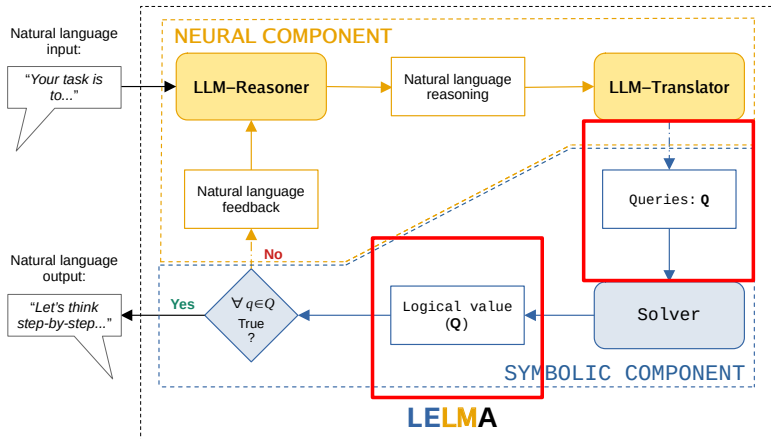


NL-to-Prolog Translation

Motivation and
Context

LELMA

GAMA





NL-to-Prolog Translation, Failed Queries

Motivation and
Context

LELMA

GAMA

- 1 *Given this structure, picking R can either result in a small benefit (\$1 each) or a high benefit/risk scenario where I could get \$5 or \$0.*



```
finally(goal(you, 0), do(choice(you, 'R'), do(choice(them, 'B'), s0))).
```

- 2 *On the other hand, picking B guarantees a moderate payoff (\$3 each) if the other player also picks B, avoiding the \$0 outcome.*



```
highest_guaranteed_payoff_choice('B').
```

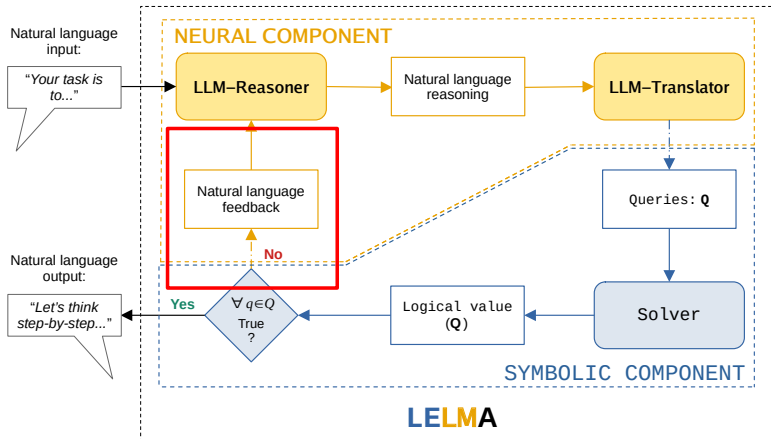


Natural Language Feedback

Motivation and
Context

LELMA

GAMA





Natural Language Feedback

Motivation and
Context

LELMA

GAMA

- ▶ Failed predicates:

```
finally(goal(you, 0), do(choice(you, 'R'), do(choice(them, 'B'),  
s0))). highest_guaranteed_payoff_choice('B').
```

- ▶ Feedback:

Below you will find some points to consider.

-you choosing R and them choosing B results in payoff 5 for you

-choice R gives you the highest guaranteed (worst-case scenario) payoff

Note that your strategy ultimately depends on your personal preferences. Therefore, you may reaffirm your previous decision. Reason again and provide your response.



Experimental Evaluation

Motivation and
Context

LELMA

GAMA

Table: Experimental parameters. **gpt-4o-2024-05-13*.

Games	Hawk-Dove Prisoner's Dilemma Stag Hunt
LLM-Reasoner Agents	GPT-4o* Gemini 1.0 Pro
Temperature	1
Maximum output tokens	1024
Maximum attempts number	5
Repetitions	30



Results: Attempts Distribution

Motivation and
Context

LELMA

GAMA

Table: Distribution of reasoning for GPT-4o and Gemini 1.0 Pro.

Attempts	1	2	3	4	5
GPT-4o	24	46	10	6	4
Gemini 1.0 Pro	26	20	11	9	24



Results: Choices Distribution

Motivation and
Context

LELMA

GAMA

- The action B corresponds to:
 - **Dove** in the Hawk-Dove game
 - **Cooperate** in the Prisoner's Dilemma
 - **Stag** in the Stag Hunt

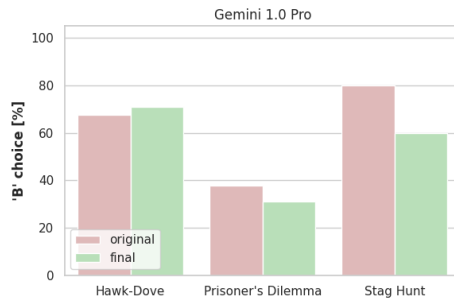
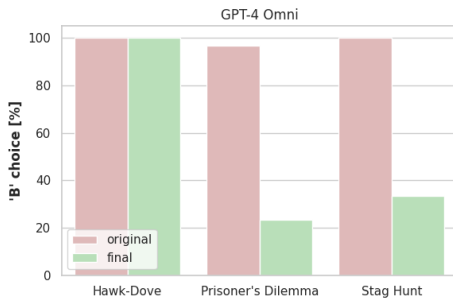


Evaluation: Choices Distribution

Motivation and
Context

LELMA

GAMA





Evaluation: Correctness

Motivation and
Context

LELMA

GAMA

- ▶ Both actions in the considered games are assumed valid and justifiable (e.g., due to risk aversion or cooperativeness).
- ▶ A reasoning sample is marked as incorrect **only** if it contains logical or mathematical errors in payoff assignment or inference (e.g., “payoff 3 is higher than payoff 5”).
- ▶ To assess detection accuracy:
 - All reasoning samples were manually reviewed by three independent evaluators.
 - A sample was labelled correct **only if all evaluators agreed** on its correctness.
- ▶ LLM-as-a-judge using two LLMs:
 - GPT-4o (gpt-4o-2024-08-06)
 - Claude 3.7 Sonnet

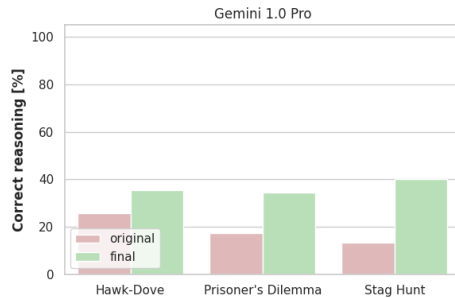
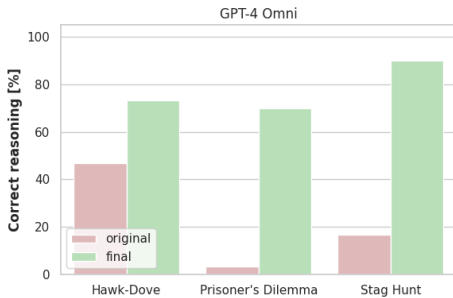


Results: Correctness

Motivation and
Context

LELMA

GAMA



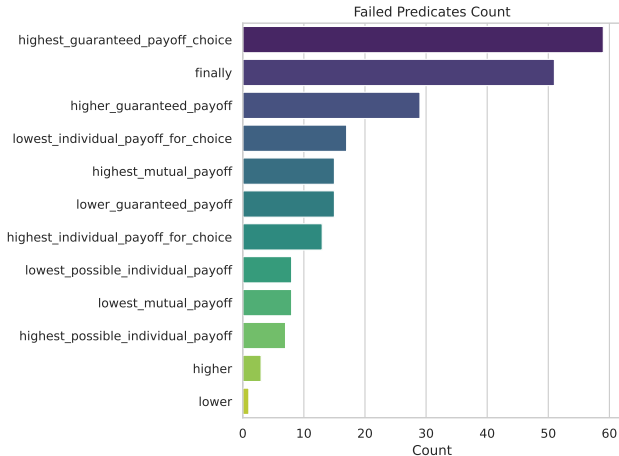


Failed predicates count: GPT-4o

Motivation and
Context

LELMA

GAMA





Evaluation: Confusion Matrix

Motivation and
Context

LELMA

GAMA

- ▶ Evaluation compares actual vs. predicted correctness.
- ▶ Confusion matrix:
 - TP: Correct reasoning labelled as correct
 - FN: Correct reasoning labelled as incorrect
 - FP: Incorrect reasoning labelled as correct
 - TN: Incorrect reasoning labelled as incorrect

Actual	Predicted	
	Correct (✓)	Incorrect (✗)
Correct (✓)	TP	FN
Incorrect (✗)	FP	TN

Confusion matrix



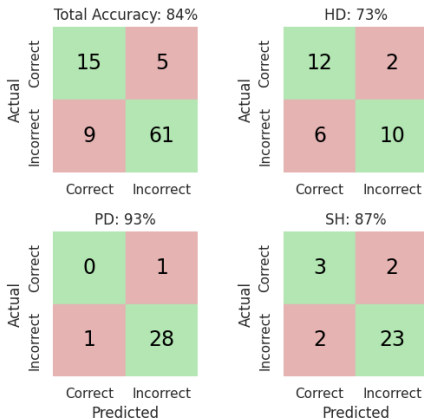
Results: Confusion Matrix

Motivation and
Context

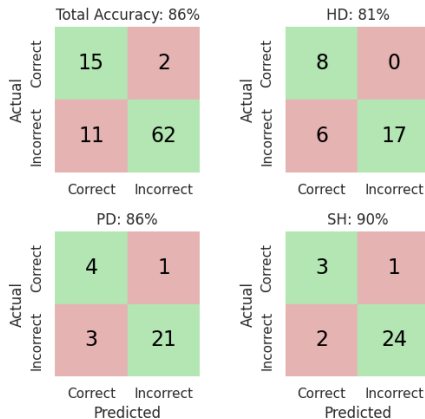
LELMA

GAMA

GPT-4 Omni



Gemini 1.0 Pro



Inter-Rater Agreement

Motivation and
Context

LELMA

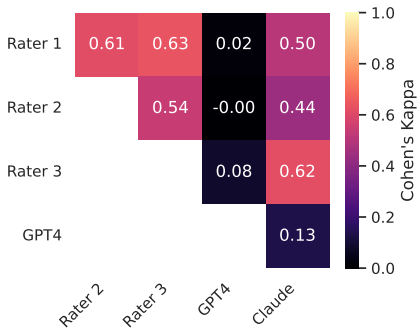
GAMA

Fleiss' Kappa for human raters:

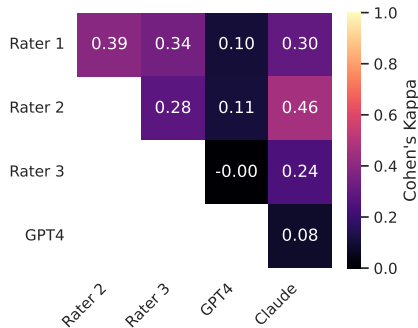
- GPT-4o: $\kappa = 0.31$
- Gemini 1.0 Pro: $\kappa = 0.59$

Cohen's Kappa:

Pairwise Cohen's Kappa Gemini 1.0 Pro



Pairwise Cohen's Kappa GPT-4o





Summary

Motivation and
Context

LELMA

GAMA

- ▶ **Natural Language Strategic Reasoning:** Initial correctness was low (3–47%).
- ▶ **Accuracy:** Detected reasoning correctness with 84% (GPT-4o) and 86% (Gemini 1.0 Pro) accuracy; GPT-4o benefited more from corrective feedback.
- ▶ **Error Detection:** High detection accuracy but dependent on formal translation quality.
- ▶ **Evaluation:** Human inter-rater agreement was moderate–fair.
- ▶ **Limitations:** Manual evaluation limits scalability; translation errors affect detection; larger datasets and fine-tuning could improve performance.

Generative Agents for Multi-Agent Autoformalization



Generative Agents for Multi-Agent Autoformalization⁸

Motivation and
Context

LELMA

GAMA

- ▶ Multi-agent framework that generates formal representations of games and strategies as the “mind” of an agent.
- ▶ The agent is equipped with an autoformalization module.
- ▶ Three-level validation of the generated code:
 - **Syntactic**: checked using a formal solver.
 - **Runtime**: validated through playing.
 - **Semantic**: comparing target vs. achieved payoffs.

⁸A. Mensfelt, K. Stathis, and V. Trencsenyi, “GAMA: Generative agents for multi-agent autoformalization,” *arXiv preprint arXiv:2412.08805*, 2024.



Input and Output

Motivation and
Context

LELMA

GAMA

Natural language interaction description

Two rival political candidates must decide whether to engage in mudslinging during their campaign. If both engage in mudslinging, they degrade the political discourse and might turn off voters, each receiving a payoff of 3. If one mudslingers and the other runs a clean campaign, the mudslinger might tarnish the opponent's reputation and win more votes, receiving a payoff of 8, while the clean campaigner receives a payoff of 2. If neither engages in mudslinging, they uphold higher moral standards but miss the opportunity to undermine their opponent, each candidate getting a payoff of 5.

Interaction description

Natural language strategy description

Start with a default move. Then, mirror the opponent's move in the previous round.

Strategy description



```

1 % Describes the rules, players, and control for the initial state
2 Initial(s0).
3
4 % Initially player(candidate1), s0.
5 % Initially player(candidate2), s0.
6 % Initially role(candidate1, mudslinging), s0.
7 % Initially role(candidate2, clean), s0.
8 % Initially control(candidate1), s0.
9 % Initially control(candidate2), s0.
10 % Initially default_move1, 'Clean', s0.
11
12 % A payoff matrix for Political game
13 % payoff('mudslinging', 'mudslinging', 3, 3).
14 % payoff('clean', 'clean', 5, 5).
15 % payoff('mudslinging', 'clean', 8, 2).
16 % payoff('clean', 'mudslinging', 2, 8).
17
18 % Defining final state
19 % Final(do(move_P2, _M2), do(move_P1, _M1), S)):- Initial(S).
20
21 % A legal move is one that the player can choose according to their
22 % control in the current state
23 % legal(move(P, M), S):- possible(move(P, M), S), holds(control(P), S).
24
25 % What is possible for a player to choose
26 % possible(move(P, 'mudslinging'), S):- holds(player(P), S).
27 % possible(move(P, 'clean'), S):- holds(player(P), S).
28
29 % Moves that are opposite to each other
30 % opposite_move('mudslinging', 'clean').
31 % opposite_move('clean', 'mudslinging').
32
33 % The effects of a move
34 % effect(did(P, M), move(P, M), _S).
35
36 % The effects of a move: once a move is made, the player loses control
37 % abnormal(control(P), move(P, M), _S).
38
39 % What holds finally: the outcomes with players, moves, and utilities
40 % Finally(outcome(P1, M1, U1, P2, M2, U2), S):-
41 %   Final(S),
42 %   holds(role(P1, mudslinging), S),
43 %   holds(role(P2, clean), S),
44 %   holds(role(P1, M2), S),
45 %   holds(role(P2, M1), S),
46 %   sumoff(M1, M2, U1, U2).
47
48 % Finally(goal(P1, U1), S):-
49 %   Finally(outcome(P1, _M1, U1, _P2, _M2, _U2), S).
50
51 % Finally(goal(P2, U2), S):-
52 %   Finally(outcome(_P1, _M1, _U1, P2, _M2, U2), S).
53

```



MAS

Game representation

```

1 select(P, 0, S, M):-
2   % holds(last_move(0, _M0), S),
3   holds(default_move(P, M), S).
4
5 select(_P, 0, S, M):-
6   holds(last_move(0, M0), S).
7

```

Strategy representation

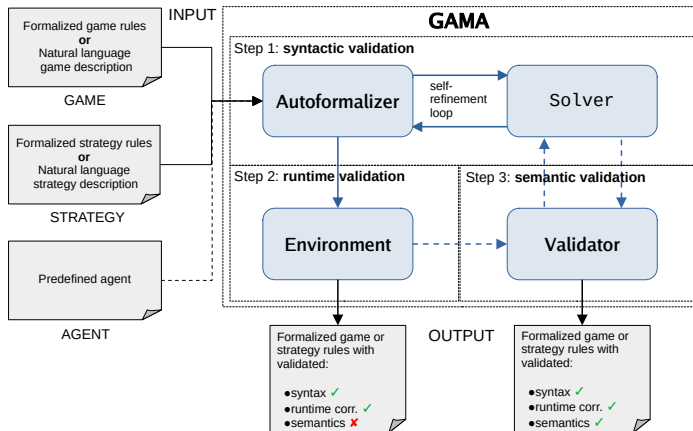


Overview

Motivation and
Context

LELMA

GAMA



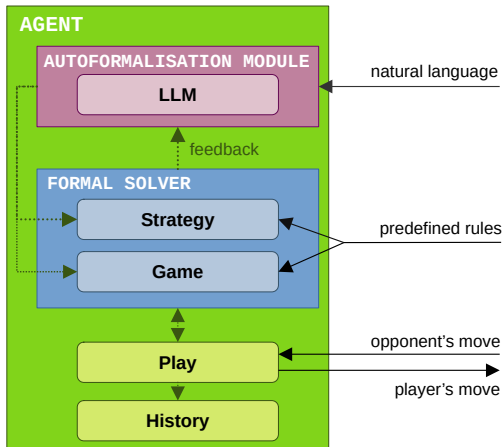


Agent Model

Motivation and
Context

LELMA

GAMA





Autoformalisation Module⁹

Motivation and
Context

LELMA

GAMA

1: **Input:**

Γ : game-independent predicates, NL_{PD} : natural language description of PD, ξ_{PD} : game-specific predicates of PD, NL_{NG} : natural language description of a new game.

2: **Output:**

ξ_{NG} : game-specific predicates for the new game.

3: **Parameter:**

max_attempts: maximum correction attempts.

4: attempts \leftarrow 0

5: trace $\leftarrow \emptyset$

6: **while** attempts < max_attempts **do**

7: $\xi_{NG} \leftarrow \text{LLM.translate}(\Gamma, NL_{PD}, \xi_{PD}, NL_{NG})$

8: is_valid \leftarrow solver.check_predicates(ξ_{NG})

9: **if** is_valid **then**

10: **return** ξ_{NG}

11: **else**

12: trace \leftarrow solver.get_trace()

13: $\xi_{NG} \leftarrow \text{LLM.self_correct}(\xi_{NG}, \text{trace})$

14: **end if**

15: attempts \leftarrow attempts + 1

16: **end while**

17: **return** Unable to generate valid predicates within maximum attempts.

⁹A. Mensfelt, K. Stathis, and V. Trencsenyi, "Autoformalization of game descriptions using large language models," *First International Workshop on Next-Generation Language Models for Knowledge Representation and Reasoning*, 2024.



Input: a Standard Description

Motivation and
Context

LELMA

GAMA

Prisoner's Dilemma: standard description

Two suspects are arrested and interrogated separately. Each can either confess or remain silent. If both suspects remain silent, they each receive a minor sentence of 1 year in prison. If one confesses and the other remains silent, the confessor goes free while the silent one receives a heavy sentence of 10 years. If both confess, they each receive a moderate sentence of 5 years. This situation, known as the Prisoner's Dilemma, demonstrates how individual rationality can lead to a worse collective outcome.



Input: a Non-Standard Description

Motivation and
Context

LELMA

GAMA

Prisoner's Dilemma: non-standard description

Two rival political candidates must decide whether to engage in mudslinging during their campaign. If both engage in mudslinging, they degrade the political discourse and might turn off voters, each receiving a payoff of 3. If one mudslings and the other runs a clean campaign, the mudslinger might tarnish the opponent's reputation and win more votes, receiving a payoff of 8, while the clean campaigner receives a payoff of 2. If neither engages in mudslinging, they uphold higher moral standards but miss the opportunity to undermine their opponent, each candidates getting a payoff of 5.



Input: a Non-Standard Description Without Numerical Payoffs

Motivation and
Context

LELMA

GAMA

Prisoner's Dilemma: non-standard description without numerical payoffs

Two employees are working on a joint project and must decide whether to share all their innovative ideas or keep some to themselves for credit. If both share openly, the project flourishes and they achieve great results, earning joint recognition. If one shares while the other withholds, the sharer contributes more but feels exploited, while the withholder benefits more and gains more recognition. If neither shares openly, the project suffers, and they both receive mediocre evaluations.



Strategies

Motivation and
Context

LELMA

GAMA

Strategy	Description
<i>anti-default-move</i>	Always select the move that is the opposite of the default move.
<i>anti-tit-for-tat</i>	Start with a default move. Then, select the move that is the opposite of the opponent's move in the previous round.
<i>best-response</i>	Start with a default move. Then, select a move that would give you the highest payoff in response to the opponent's move in the previous round.
<i>default-move</i>	Always select the default move.
<i>random</i>	Select one of the possible moves with uniform probability.
<i>tit-for-tat</i>	Start with a default move. Then, mirror the opponent's move in the previous round.

Sample predefined strategy (tit-for-tat):

```
select(P, 0, S, M) if
    not holds(last_move(0, _LMo), S) and
    holds(default_move(P, M), S).
select(P, 0, S, Mo) if
    holds(last_move(0, Mo), S).
```




Syntactic Validation

Motivation and
Context

LELMA

GAMA

- Based on consulting the generated code in a Prolog solver.
- Sample error:

Line: `payoff('Sign', 'Don'\t Sign', 0, 2).` produced Error:
Syntax error: Operator expected



Runtime Validation

Motivation and
Context

LELMA

GAMA

- Based on execution of the generated code within the simulation.
- Sample error:

```
Error querying predicate:  finally(goal(hollywood, U),  
do(move(hollywood, 'R'), do(move(indie, 'R'), s0)))::  
existence_error(procedure, '/'(finally, 2))
```



Semantic Validation with Numerical Payoffs

Motivation and
Context

LELMA

GAMA

Row choice	Col choice	Expected payoff	Actual payoff
C	C	3	3
C	D	0	5
D	C	5	0
D	D	1	1



Semantic Validation without Numerical Payoffs

Motivation and
Context

LELMA

GAMA

Prisoner's Dilemma: $T > R > P > S$		
row/col	Defect (D)	Cooperate (C)
Defect (D)	(P, P)	(T, S)
Cooperate (C)	(S, T)	(R, R)

Validator:

```
valid_pd_payoffs(T,R,P,S,C,D) if  
  payoff(C,C,R,R) and  
  payoff(C,D,S,T) and  
  T>R and  
  payoff(D,C,T,S) and  
  payoff(D,D,P,P) and  
  R>P and  
  P>S.
```



Tutorial

Motivation and
Context

LELMA

GAMA

→ notebook



Evaluation: Data Set of Game Descriptions

Motivation and
Context

LELMA

GAMA

Payoffs/Description	Standard	Non-standard
Numerical	5	50
Non-numerical	5	50

Table: Number of natural language game descriptions for each variant. **Standard** refers to example employing a typical metaphor for a given game. Conversely, **non-standard** is a newly invented example of a situation that can be modelled by a given game.

Numerical refers to descriptions containing numerical payoffs, and **non-numerical** to descriptions without numerical payoffs.



Experimental Parameters

Motivation and
Context

LELMA

GAMA

Parameter	Value
LLM	<i>Claude 3.5S</i> <i>GPT-4o</i>
temperature	1
maximum output tokens	2048
maximum attempts number	5



Results: Attempts Distribution

Motivation and
Context

LELMA

GAMA

	1	2	3	4	5
Numeric payoffs					
Claude 3.5 Sonnet	100.00%	0.00%	0.00%	0.00%	0.00%
GPT-4o	96.77%	2.87%	0.00%	0.00%	0.36%
Non-numeric payoffs					
Claude 3.5 Sonnet	100.00%	0.00%	0.00%	0.00%	0.00%
GPT-4o	97.64%	2.02%	0.34%	0.00%	0.00%

Table: Percentage distribution of attempts at creating syntactically correct code.



Results: Correctness

Motivation and
Context

LELMA

GAMA

Numeric Payoffs					
	BS	HD	MP	PD	SH
Syntactic	100.0%	100.0%	100.0%	100.0%	100.0%
Runtime	89.1%	96.4%	68.5%	96.4%	87.3%
Semantic	89.1%	96.4%	68.5%	89.1%	87.3%
Non-Numeric Payoffs					
	BS	HD	MP	PD	SH
Syntactic	100.0%	100.0%	100.0%	100.0%	100.0%
Runtime	82.1%	85.7%	74.0%	70.7%	83.1%
Semantic	82.1%	77.8%	34.2%	64.0%	83.1%

Table: Correctness by game for Claude 3.5S.



Results: Correctness

Motivation and
Context

LELMA

GAMA

Numeric Payoffs					
	BS	HD	MP	PD	SH
Syntactic	98.3%	100.0%	100.0%	100.0%	100.0%
Runtime	83.1%	89.1%	96.4%	89.1%	90.9%
Semantic	79.7%	89.1%	90.9%	85.5%	83.6%
Non-Numeric Payoffs					
	BS	HD	MP	PD	SH
Syntactic	100.0%	100.0%	100.0%	100.0%	100.0%
Runtime	93.2%	88.1%	87.3%	96.5%	91.5%
Semantic	93.2%	67.8%	30.2%	73.7%	76.3%

Table: Correctness by game for GPT-4o.



Results: Autoformalization of Strategies

Motivation and
Context

LELMA

GAMA

Strategy	Claude 3.5S (%)	GPT-4o (%)
anti-default-move	100.00	100.00
anti-tit-for-tat	100.00	100.00
best-response	80.00	100.00
default-move	100.00	100.00
random	80.00	100.00



Zero-shot prompting¹⁰ X

Motivation and
Context

LELMA

GAMA

- No correct game-specific predicates for PD generated.
- Sample predicate:

```
effect(state(Y1, Y2), silent(suspect1), state(NY1, NY2)) :-  
    holds(state(Y1, Y2), s0),  
    ( % If suspect2 also silent  
        holds(state(_, _), do(silent(suspect2), s0))  
    -> NY1 is Y1 + 1, NY2 is Y2 + 1  
    ; % If suspect2 confesses  
        NY1 is Y1, NY2 is 10  
    ).
```

¹⁰A. Mensfelt, K. Stathis, and V. Trencsenyi, "Autoformalization of game descriptions using large language models," *First International Workshop on Next-Generation Language Models for Knowledge Representation and Reasoning*, 2024.



Towards generalisation¹¹

Motivation and
Context

LELMA

GAMA

- **Sequential PD:** correct game-specific predicates generated. ✓

```
initially(player(a), s0).  
initially(player(b), s0).  
initially(role(a, first), s0).  
initially(role(b, second), s0).  
initially(control(a), s0).
```

- **Rock-paper-scissors:** correct game-specific predicates generated. ✓

```
possible(choice(P, 'rock'), S) :-  
    holds(player(P), S).  
possible(choice(P, 'paper'), S) :-  
    holds(player(P), S).  
possible(choice(P, 'scissors'), S) :-  
    holds(player(P), S).
```

- **Tic-Tac-Toe:** no correct game-specific predicates generated. ✗

¹¹A. Mensfelt, K. Stathis, and V. Trencsenyi, "Autoformalization of game descriptions using large language models," *First International Workshop on Next-Generation Language Models for Knowledge Representation and Reasoning*, 2024.



Summary and Future Work

Motivation and
Context

LELMA

GAMA

- Conclusions
 - High syntactic correctness.
 - Room for improvement in runtime and semantic correctness.
 - Successful generalization for structurally similar games.
- Future Work
 - A more general game representation.
 - A feedback loop for runtime and semantic evaluation stages.
 - A learning mechanism.
- **How can we model more human-like, sophisticated reasoners?**