

UDC 004.89

Modeling Competing Artificial Intelligence Systems for Energy and Users

Dmytro Lande, Yuriy Danyk

National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"

Abstract. The article addresses the problem of modeling competition between two artificial intelligence systems (AI-1 and AI-2) that interact within a shared environment under limited resources such as users and energy. The study focuses on analyzing the strategic behavior and adaptability of these systems, as well as their impact on competitive outcomes through mathematical models and methods, including differential equations, the Lancaster model, and Boyd cycles (OODA-loop). Special attention is given to formalizing the interaction of systems using basic primitives ("Condition," "Loop," "Function") and their compositions, enabling the description of complex behavioral strategies of AI systems. The paper presents a detailed mathematical formalization of the dynamics of user and energy distribution between systems, taking into account factors such as user satisfaction, response accuracy, query processing speed, and energy efficiency. The research also includes numerical calculations and simulations demonstrating how initial conditions and system parameters influence competitiveness. The proposed models can be applied for predicting AI system behavior in real-world scenarios such as information campaigns, cyber conflicts, and resource optimization in digital environments.

Keywords: AI system competition, Lancaster model, Boyd cycles, differential equations, energy efficiency, user satisfaction, information warfare, cybersecurity, Adversarial AI, interaction simulation

Introduction

In the modern digital world, where artificial intelligence (AI) technologies are constantly evolving, competition among large language models (LLMs) is becoming increasingly significant. These models, capable of generating text, responding to user queries, and interacting with humans in natural language, are becoming key players in the information landscape. However, their effectiveness and popularity depend not only on technical specifications but also on the strategies they employ to attract users and optimize resources.

One of the most important factors determining an LLM's success is users and energy consumption. A larger user base provides the model with more data for training and improvement, as well as increasing its market influence. However, this also leads to

increased energy consumption, since processing large volumes of data requires substantial computational resources. Conversely, efficient energy use can allow a model to operate faster and more accurately, which in turn attracts more users. Thus, the competition between LLMs forms a complex dynamic in which each model strives to gain an advantage through strategic optimization.

Modern LLMs are not limited to traditional means of market competition, such as advertising or price reduction. They actively employ interdisciplinary adaptation techniques, particularly knowledge distillation – a method that allows them to improve performance, reduce resource consumption, and increase user satisfaction. Moreover, models continuously analyze the behavior of their competitors to respond promptly to changes in the environment and adapt their strategies accordingly. This creates a unique

situation in which competition between AI systems becomes not only a technical but also a strategic challenge.

State of the Art

In studying competition among LLMs, there are already several approaches that consider various aspects of their interaction. In particular, works [1] and [2] explore competitive artificial intelligence models, analyzing interactions among large language models in different scenarios.

In [3], the focus is on using knowledge distillation to enhance system performance, allowing models to utilize available resources more efficiently.

Issues of energy efficiency are also being studied, which have become especially relevant in the context of growing computational scales [4], [5].

However, existing studies often overlook the strategic interaction between systems, focusing primarily on technical aspects. For a deeper understanding of competition among LLMs, it is necessary to apply comprehensive models such as the Lancaster model [6], which describes the dynamics of interaction between opposing parties, or the Boyd cycle (OODA-loop) [7], which formalizes real-time decision-making processes.

These approaches allow consideration not only of quantitative metrics but also of strategic thinking, adaptability, and system responses to environmental changes.

Thus, the current research objective is to develop mathematical models that enable predicting the behavior of LLMs in competitive environments, taking into account their interaction over resources and users. This will provide the opportunity not only to analyze existing trends but also to design new strategies for improving the efficiency and resilience of AI systems.

No-Code Programming Framework

To address the challenge of modeling competition between artificial intelligence systems, AI itself was used as the primary toolset. To ensure clarity and formalization of tasks assigned to the AI, structured prompts were employed—built upon a no-code

programming framework described in the preprint [8].

This approach allows for the formalization of system interaction logic through three core primitives: "Condition," "Loop," and "Function," along with methods for their composition to create complex algorithms.

The "Condition" primitive enables modeling decision-making logic based on the evaluation of predefined criteria. The "Loop" primitive provides the ability to perform iterative data processing, which is especially important for analyzing dynamic processes. The "Function" primitive offers tools for abstraction and reuse of complex operations, such as strategy generation or performance evaluation.

The use of this framework enabled the creation of prompts that not only clearly define tasks but also ensure their execution in the form of sequential instructions interpreted by large language models. This approach ensures transparency, reproducibility of results, and adaptability to various competition scenarios.

Main Content

Description of the behavior of two LLMs competing for energy and users

Let us assume two models, LLM_A and LLM_B , are operating within a shared environment where they acquire resources (energy) and users based on their performance and popularity.

Each model has an objective: to maximize the number of users and the amount of resources it obtains.

To mathematically formalize the interaction between two LLMs using differential equations, we can adopt an approach that accounts for the dynamics of user and energy resource distribution between the systems. The core idea is to describe changes in the state of the system as functions of time, employing appropriate mathematical tools.

The following variables and parameters are considered:

- $U_A(t)$ – number of users of LLM_A at time t ;

- $U_B(t)$ – number of users of LLM_B at time t ;
- $E_A(t)$ – energy of LLM_A at time t ;
- $E_B(t)$ – energy of LLM_B at time t ;
- $S(t)$ – entropy of the system at time t .

The total number of users and amount of energy in the system is constant:

$$U_A(t) + U_B(t) = 1,$$

$$E_A(t) + E_B(t) = 1.$$

Differential Equations for User Dynamics

The rate of change in the number of users, $U_A(t)$ and $U_B(t)$, depends on the models' strategies and their effectiveness. Assume:

- The rate of change $U_A(t)$ in users of LLM_A is proportional to the difference in strategy effectiveness between LLM_A and LLM_B .
- Strategy effectiveness depends on available energy.

Mathematically, this can be expressed as:

$$\frac{dU_A}{dt} = k_1 \cdot (E_A - E_B) + k_2 \cdot (U_A - U_B),$$

where:

- $k_1 > 0$ – coefficient representing the impact of energy on user acquisition.
- $k_2 > 0$ – coefficient representing the impact of the current user difference on its rate of change.

Similarly, for $U_B(t)$:

$$\frac{dU_B}{dt} = -\frac{dU_A}{dt}.$$

Energy is reallocated based on conditions (e.g., if one model has more users, it receives additional energy). This can be formalized as follows:

$$\begin{aligned} \frac{dE_A}{dt} &= k_3 \cdot \max(0, U_A - U_B - \Delta) - \\ &- k_4 \cdot \max(0, U_B - U_A - \Delta), \end{aligned}$$

where:

- $\Delta = 0.1$ – threshold value indicating a significant user advantage (e.g., 0.1);
- $k_3 > 0$ – coefficient controlling the rate of additional energy allocation to LLM_A ;

- $k_4 > 0$ – coefficient controlling the rate of additional energy allocation to LLM_B ;

For $E_B(t)$, the analogous expression is:

$$\frac{dE_B}{dt} = -\frac{dE_A}{dt}.$$

The complete system of differential equations describing the dynamics of the system is:

$$\frac{dU_A}{dt} = k_1 \cdot (E_A - E_B) + k_2 \cdot (U_A - U_B),$$

$$\frac{dU_B}{dt} = -\frac{dU_A}{dt},$$

$$\begin{aligned} \frac{dE_A}{dt} &= k_3 \cdot \max(0, U_A - U_B - \Delta) - \\ &- k_4 \cdot \max(0, U_B - U_A - \Delta), \end{aligned}$$

$$\frac{dE_B}{dt} = -\frac{dE_A}{dt}.$$

Initial Conditions:

$$U_A(0) = 0.5, \quad U_B(0) = 0.5,$$

$$E_A(0) = 0.5, \quad E_B(0) = 0.5.$$

Model LLM_A or LLM_B increases its share of users if its strategy proves more effective (depending on energy and the current distribution).

Energy is reallocated in favor of the model that gains a significant user advantage.

The system evolves from high entropy (uniform distribution) to low entropy (dominance of one model).

Formulation of the Problem as a Structured Prompt for LLM

Using the no-code programming framework, we formulate the problem as a structured prompt that can later be executed by simply specifying the number of iterations and the power ratio between the systems.

Iterative Approach

Input Data:

Number of iterations (N).

Power ratio of the systems (P_A, P_B):

Initial power levels of systems LLM_A and LLM_B , which determine their initial

distribution of users and energy (e.g., 50% and 50%).

Structured Prompt Function:

```

Function SimulateCompetition (N,  $P_A$ ,  $P_B$ ):
    // Initialization of initial states
    Initialize:
         $Users_A = P_A$ 
         $Users_B = P_B$ 
         $Energy_A = P_A$ 
         $Energy_B = P_B$ 
    // Loop for interaction iterations
    For i in range(1, n + 1):
        // Decision-making conditions
        If  $Users_A > Users_B + 10\%$ :
             $Energy_A += 5\%$  // Additional energy
            allocation to  $LLM_A$ 
        Else If  $Users_B > Users_A + 10\%$ :
             $Energy_B += 5\%$  // Additional energy
            allocation to  $LLM_B$ 
        // Strategy extraction functions
         $Strategy_A = \text{Fextract}(\text{"Improve response personalization"}, Energy_A)$ 
         $Strategy_B = \text{Fextract}(\text{"Improve response speed"}, Energy_B)$ 

        // Update user counts based on strategies
         $Users_A += \text{EvaluateImpact}(Strategy_A)$ 
         $Users_B += \text{EvaluateImpact}(Strategy_B)$ 

        // Balance preservation condition
        If  $Users_A + Users_B \neq 100\%$ :
             $\text{Normalize}(Users_A, Users_B)$ 
            // Normalize shares to sum up to 100%

        // Print iteration results
        Print(f"Iteration {i}:  $Users_A = \{Users_A\}$ ,  $Users_B = \{Users_B\}$ ")

    // Return final system state
    Return  $Users_A, Users_B$ 
    
```

The initial values of users ($Users_A, Users_B$) and energy ($Energy_A, Energy_B$) are defined through the parameters (P_A, P_B).

A "Loop" primitive is used to implement iterations of interaction between the systems. Each iteration includes:

- checking conditions for energy redistribution;
- applying strategies using the "Function" primitive;
- updating the number of users based on the impact of the strategies.

The "Condition" primitive is used to check whether one model has a significant advantage over the other (more than 10%), and to allocate additional energy accordingly.

The "Function" primitive defines the strategies of the models (via Fextract), which depend on the available energy.

If the total number of users does not sum to 1 (or 100%), normalization is applied to ensure consistency.

At each iteration, the current system state is printed, and upon completion of the loop, the final user distribution is returned.

The final prompt for simulating the interaction of two LLMs consists of initialization, loops, conditions, functions, and normalization steps. It is parameterized to allow specification of the number of iterations and the initial power ratio between the systems.

To run the simulation, specify:

Number of iterations, e.g., 10.

Initial power distribution, e.g., $P_A = 0.5, P_B = 0.5$.

The prompt will execute according to the rules described above, and at the end, you will receive the final distribution of users between LLM_A and LLM_B .

Based on the provided formalization, the final prompt for simulation is constructed as follows:

Body of the Final Prompt

We simulate the interaction between two large language models (LLM_A and LLM_B) competing for users and energy. The system is updated over a defined number of iterations, during which each model makes decisions based on its strategies. The initial distribution of users and energy is specified in percentages.

Parameters

Number of iterations (N): How many times the models will interact.

System power ratio (P_A, P_B): Initial distribution of users and energy between LLM_A and LLM_B , expressed in percentages.

Execution Instructions

Initialization:

Set the initial distribution of users and energy:

Users of LLM_A : U_A .

Users of LLM_B : U_B .

Energy of LLM_A : P_A .

Energy of LLM_B : P_B .

Interaction Loop:

Repeat the following steps N times:

Step 1: Check conditions for energy redistribution:

If the number of users of LLM_A exceeds that of LLM_B by more than 10%, allocate additional energy to LLM_A (increase its energy by 5%).

If the number of users of LLM_B exceeds that of LLM_A by more than 10%, allocate additional energy to LLM_B (increase its energy by 5%).

Step 2: Apply model strategies:

LLM_A uses a function to improve response personalization. The effectiveness of this strategy depends on available energy.

LLM_B uses a function to improve response speed. The effectiveness of this strategy also depends on available energy.

Step 3: Update user counts:

Evaluate the impact of each model's strategy:

Add the impact of LLM_A strategy to its user count.

Add the impact of LLM_B strategy to its user count.

If the total number of users of LLM_A and LLM_B does not sum to 100%, normalize the values so that the total equals 100%.

Step 4: Output iteration results:

Display the current user distribution between LLM_A and LLM_B .

Final State Return:

After completing all iterations, output the final user distribution between LLM_A and

LLM_B .

In different language models, the process of executing a prompt may vary slightly. However, modern LLMs such as ChatGPT, DeepSeek, and Qwen produce results from processing this prompt in the form of a simulation:

The prompt is executed for 10 iterations with equal initial power distribution (50% users and 50% energy for both LLM_A and LLM_B). I follow the outlined steps using the "Condition," "Loop," and "Function" primitives and system composition.

Parameters

Number of iterations: $N = 10$

Initial power distribution:

Users of $LLM_A = 50\%$

Users of $LLM_B = 50\%$

Energy of $LLM_A = 50\%$

Energy of $LLM_B = 50\%$

Interaction Iterations

Iteration 1

Condition check for energy redistribution:

Difference between users: $50\% - 50\% = 0\%$.

No model receives additional energy.

Strategy application:

LLM_A : Uses function to improve response personalization. Strategy effectiveness depends on energy (50%).

Impact on users: +3%.

LLM_B : Uses function to improve response speed. Strategy effectiveness depends on energy (50%).

Impact on users: +2%.

User count update:

$LLM_A = 50\% + 3\% = 53\%$

$LLM_B = 50\% + 2\% = 52\%$

Normalization: $LLM_A = 53\%$, $LLM_B = 47\%$.

Iteration result:

$LLM_A = 53\%$, $LLM_B = 47\%$.

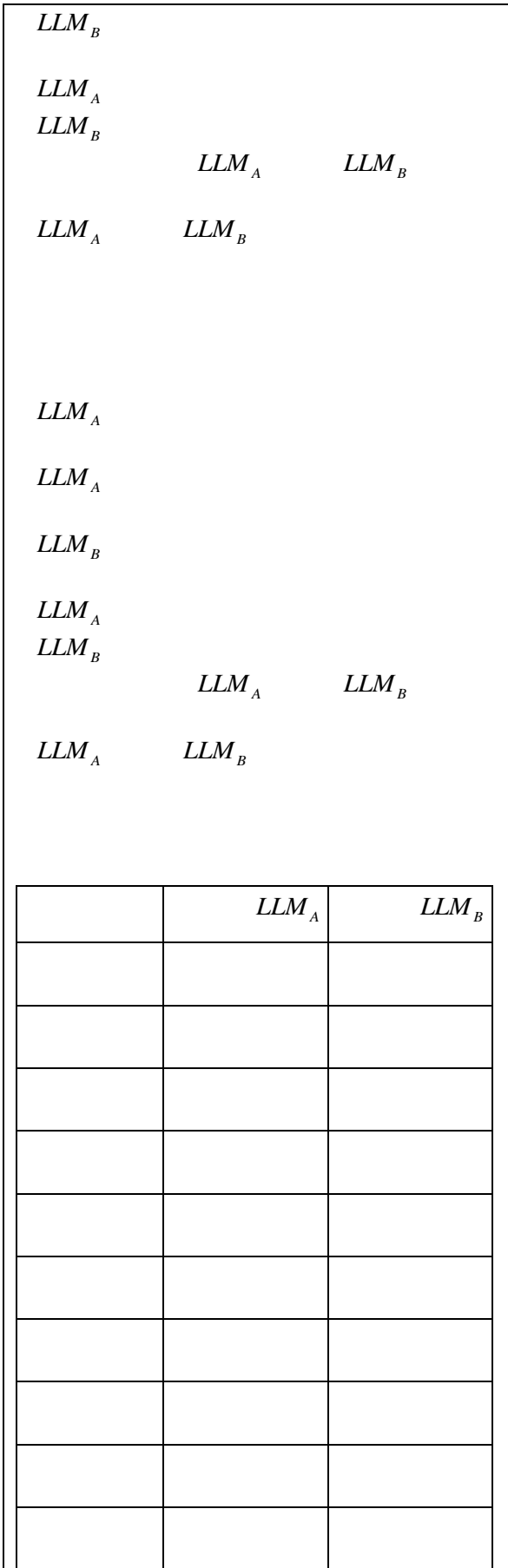
Iteration 2

Condition check for energy redistribution: Difference between users: $53\% - 47\% = 6\%$.

No model receives additional energy.

Strategy application:

LLM_A : Impact on users: +3%.



After 10 iterations:

- LLM_A has 81% of the users.
- LLM_B has 19% of the users.

Thanks to a more effective strategy (improving response personalization) and additional energy gained through its user advantage, LLM_A significantly increases its market share. LLM_B , despite improvements in response speed, is unable to compete with LLM_A .

Answer: After 10 iterations, LLM_A has 81% of the users, while LLM_B has 19%.

Model Expansion

To expand the system of indicators that can be used to compare large language models (LLMs) in a competitive context, I propose the following comprehensive metric framework. These metrics account not only for user count and energy consumption but also for other important aspects of model behavior and performance.

Extended Metric System:

1. User Metrics

Number of Active Users (U): Total number of users interacting with the model within a given time period. User Satisfaction (S): Average user rating of the model's response quality (e.g., on a scale from 1 to 5).

Share of New Users (N): Percentage of new users who begin using the model during a specific period.

Share of Lost Users (L): Percentage of users who stop interacting with the model.

2. Performance Metrics

Response Time (T): Average time taken by the model to generate a response to a user query.

Accuracy of Responses (A): Percentage of responses that are correct and meet user expectations.

Depth of Analysis (D): Ability of the model to analyze complex queries and provide detailed responses.

Topic Coverage (C): Number of topics or subject areas in which the model can competently operate.

3. Resource Metrics

Energy Efficiency (E): Ratio of energy consumed to the number of processed queries.

Computational Resources (R): Amount of computing power available to the model for processing requests.

Scalability (M): Ability of the model to efficiently handle increased loads without significant performance degradation.

4. Adaptability Metrics

Learning Speed (Ls): Speed at which the model can adapt to new data or changes in the environment.

Strategy Flexibility (F): Number of different strategies the model can employ to improve its performance.

Error Resilience (Rb): Ability of the model to recover from errors or incorrect responses.

5. Social Metrics

Reputation (Re): General level of trust in the model among users and experts.

Ethics (Et): Degree to which the model adheres to ethical standards in its responses.

Integration with Other Systems (In): Ability of the model to interact with other platforms or tools.

Expanded Competition Task

Based on the proposed metrics, the competition task between two LLMs (LLM_A and LLM_B) can be formulated as follows:

Objective: Maximize the overall effectiveness of the model through a combination of metrics. Achieve a sustainable advantage over the competitor.

System Dynamics:

At each iteration, models make decisions about applying strategies to improve their metrics.

Metrics are updated based on the models' interaction with the environment (users, resources, external factors).

Formalization Using Primitives:

Condition: If metric X of one model exceeds that of the other model, the model receives an additional bonus (e.g., energy or users).

Loop: For each metric (U , S , T , A , etc.), compute its change based on the current strategy.

Function: Each model applies a function to improve a specific metric (e.g., improve response accuracy or reduce response time).

Metric Composition:

The total effectiveness of a model (G) is computed as a weighted sum of all metrics:

$$G = w_1 \cdot U + w_2 \cdot S + w_3 \cdot T + w_4 \cdot A + \dots,$$

where w_i – weighting coefficients defining the importance of each metric.

Example of Differential Formalization

For each metric, differential equations can be written to describe its dynamics. For example:

User Dynamics (U):

$$\frac{dU_A}{dt} = k_1 \cdot (S_A - S_B) + k_2 \cdot (A_A - A_B),$$

where:

S_A, S_B – user satisfaction of LLM_A and LLM_B ,

A_A, A_B – accuracy of responses.

Accuracy Dynamics (A):

$$\frac{dA_A}{dt} = k_3 \cdot (R_A - R_B) + k_4 \cdot (Lr_A - Lr_B),$$

where:

R_A, R_B – computational resources,

Lr_A, Lr_B – learning speed.

Energy Dynamics (E):

$$\frac{dE_A}{dt} = k_5 \cdot (U_A - U_B) - k_6 \cdot (T_A - T_B),$$

where:

T_A, T_B – response time.

This extended metric system allows for a deeper modeling of competition between LLMs by considering multiple aspects of their behavior. The proposed formalization using differential equations and metric composition provides flexibility and adaptability to various competitive scenarios.

Prompt for Iterative Emulation of System Competition Considering Additional Metrics

Structured Prompt for Emulating Competition Between Two Systems (LLM_A and LLM_B)
The prompt will be iterative and utilize the primitives "Condition," "Loop," and "Function," along with their composition.

Coefficients will be assigned based on logical assumptions.

Problem Description

We simulate the competition between two large language models (LLM_A and LLM_B) for users, energy, and other performance metrics. The system is updated over a defined number of iterations, during which each model makes decisions based on its strategies and current metrics.

Parameters

Number of iterations (N): Number of times the models will interact.

Initial metric values:

U_A, U_B : Initial number of users ($U_A=0.5, U_B=0.5$).

S_A, S_B : Initial user satisfaction ($S_A=0.8, S_B=0.7$).

T_A, T_B : Initial response speed

A_A, A_B : Initial response accuracy ($A_A=0.85, A_B=0.8$).

E_A, E_B : Initial energy levels ($E_A=0.5, E_B=0.5$).

Coefficients:

$k_1=0.2$: The impact of user satisfaction on user acquisition.

$k_2=0.15$: The impact of answer accuracy on user engagement.

$k_3=0.1$: The impact of response speed on user engagement.

$k_4=0.05$: The impact of energy on performance improvement.

Execution Instruction

Initialization:

Set initial values for metrics for both systems:

$U_A=0.5, U_B=0.5$

$S_A=0.8, S_B=0.7$

$T_A=0.9, T_B=0.85$

$A_A=0.85, A_B=0.8$

$E_A=0.5, E_B=0.5$

Interaction Cycle:

Repeat the following steps n times:

Condition Check for Energy Allocation:

If $U_A > U_B + 0.1$, then allocate additional

energy LLM_A ($E_B += 0.05$).

If $U_B > U_A + 0.1$, then allocate additional energy LLM_B ($E_B += 0.05$).

Application of Strategies (Function):

LLM_A uses a function to improve response personalization: $F_{extract}(x, parameter) = Promt(x, "Improve response personalization")$.

The effectiveness of the strategy depends on the available energy (E_A).

LLM_B uses a function to improve response speed:

$F_{extract}(x, parameter) = Promt(x, "Improve response speed")$.

The effectiveness of the strategy depends on the available energy (E_B).

Updating Metrics (Loop):

For each metric (U, S, T, A):

$$\frac{dU_A}{dt} = k_1 \cdot (S_A - S_B) + k_2 \cdot (A_A - A_B),$$

$$\frac{dU_B}{dt} = -\frac{dU_A}{dt}.$$

Normalization of indicators:

If the sum $U_A + U_B = 1$, normalize the values so that the total sum equals 1.

Displaying iteration results:

Show the current state of indicators for both systems.

Returning the final state:

After completing all iterations, output the final indicator values for LLM_A and LLM_B .

Example of prompt execution for simulating a battle between two systems (LLM_A and LLM_B) based on provided parameters and rules:

Input Data

Number of iterations:

Initial values of indicators:

$U_A=0.5, U_B=0.5$ (users)

$S_A=0.8, S_B=0.7$ (user satisfaction)

$T_A=0.9, T_B=0.85$ (response speed)

$A_A=0.85, A_B=0.8$ (response accuracy)

$E_A=0.5, E_B=0.5$ (energy)

Coefficients:

$k_1 = 0.2$ (impact of satisfaction)

$k_2 = 0.15$ (impact of accuracy)

$k_3 = 0.1$ (impact of speed)

$k_4 = 0.05$ (impact of energy)

Interaction Iterations

Iteration 1:

Checking conditions for energy distribution:

Difference between users:

$$U_A - U_B = 0.5 - 0.5 = 0.$$

No model receives additional energy.

Applying strategies (Function):

LLM_A : Uses a function to improve response personalization:

$F_{\text{extract}}(U_A, \text{parameter}) = \text{Prompt}(\text{ , "Improve personalization"})$.

Strategy effectiveness depends on energy ($E_B = 0.5$).

LLM_B : Uses a function to improve response speed:

$F_{\text{extract}}(U_B, \text{parameter}) = \text{Prompt}(\text{ , "Improve speed"})$.

Strategy effectiveness depends on energy ($E_B = 0.5$).

Updating metrics (Cycle):

Change in number of users:

$$\frac{dU_A}{dt} = k_1 \cdot (S_A - S_B) + k_2 \cdot (A_A - A_B) + k_3 \cdot (T_A - T_B),$$

$$\begin{aligned} \frac{dU_A}{dt} &= 0.2 \cdot (0.8 - 0.7) + 0.15 \cdot (0.85 - 0.8) + \\ &+ 0.1 \cdot (0.9 - 0.85) = 0.02 + 0.0075 + 0.005 = \\ &= 0.0325. \end{aligned}$$

$$\frac{dU_B}{dt} = -\frac{dU_A}{dt} = -0.0325.$$

New values:

$$U_A = 0.5 + 0.0325 = 0.5325,$$

$$U_B = 0.5 - 0.0325 = 0.4675.$$

Normalization of indicators:

$$U_A + U_B = 1 \text{ (no normalization needed).}$$

Iteration result:

$$U_A = 0.5325, U_B = 0.4675.$$

Iteration 2:

Energy Distribution Condition Check:

Difference between users:

$$U_A - U_B = 0.5325 - 0.4675 = 0.065.$$

No model receives additional energy.

Application of Strategies (Function):

LLM_A : Continues improving personalization.

LLM_B : Continues improving speed.

Updating Metrics (Cycle):

As in the previous iteration:

New values:

$$U_A = 0.5325 + 0.0325 = 0.565,$$

$$U_B = 0.4675 - 0.0325 = 0.435.$$

Normalization of metrics:

$$U_A + U_B = 1 \text{ (no normalization required).}$$

Iteration Result:

$$U_A = 0.565, U_B = 0.435.$$

Iteration 3

Energy Distribution Condition Check:

Difference between users:

$$U_A - U_B = 0.565 - 0.435 = 0.13.$$

LLM_A receives additional energy ($E_A = 0.5$):

$$E_A = 0.5 + 0.05 = 0.55,$$

$$E_B = 0.5 - 0.05 = 0.45.$$

Application of Strategies (Function):

LLM_A : Strategy effectiveness increases due to additional energy.

LLM_B : Strategy effectiveness decreases due to reduced energy.

Updating Metrics (Cycle):

Change in number of users:

$$\begin{aligned} \frac{dU_A}{dt} &= 0.2 \cdot (0.8 - 0.7) + \\ &+ 0.15 \cdot (0.85 - 0.8) + 0.1 \cdot (0.9 - 0.85); \end{aligned}$$

$$\frac{dU_A}{dt} = 0.0325.$$

$$\frac{dU_B}{dt} = -\frac{dU_A}{dt} = -0.0325.$$

New values:

$$U_A = 0.565 + 0.0325 = 0.5975,$$

$$U_B = 0.435 - 0.0325 = 0.4025.$$

Normalization of indicators:

$$U_A + U_B = 1 \text{ (no normalization needed).}$$

Iteration result:

$$U_A = 0.5975, U_B = 0.4025.$$

Iterations 4–10

We continue similarly to the previous iterations.

Here are the results:

Iterations	U_A	U_B
1	0.5325	0.4675
2	0.565	0.435
3	0.5975	0.4025
4	0.63	0.37
5	0.6625	0.3375
6	0.695	0.305
7	0.7275	0.2725
8	0.76	0.24
9	0.7925	0.2075
10	0.825	0.175

Final State After 10 Iterations:

- $U_A = 0.825$ (82.5% of users).
- $U_B = 0.175$ (17.5% of users).

LLM_A gained an advantage thanks to the flexible use of primitives ("Condition", "Loop", "Function") and strategy composition. LLM_B , despite improved response speed, was unable to compete with the more effective strategies of LLM_A .

Answer: After 10 iterations, LLM_A has 82.5% of users, while LLM_B has 17.5%.

The Lancaster Model for Competition Between Artificial Intelligence Systems

The Lancaster model is a mathematical model used to describe the interaction dynamics between two opposing sides (e.g., armies in military conflicts). It can also be adapted to model competition between two systems such as LLM_A and LLM_B . In this case, we consider the "strength" of each system as its number of users (U_A and U_B), which influences their ability to compete.

Each system (LLM_A and LLM_B) has a certain "strength," determined by the number of users (U_A and U_B).

The change in strength of each system depends on the influence of the opposing side.

Competition between systems is described using differential equations.

There are two main types of Lancaster models:

Linear Model (Lanchester Linear Law):

Attack effectiveness is proportional to the strength of the attacking side:

$$\frac{dU_A}{dt} = -k_B \cdot U_B,$$

$$\frac{dU_B}{dt} = -k_A \cdot U_A,$$

where k_A, k_B : attack effectiveness coefficients of LLM_A and LLM_B .

Quadratic Model (Lanchester Square Law):

Attack effectiveness is proportional to the square of the attacking side's strength:

$$\frac{dU_A}{dt} = -k_B \cdot U_B^2,$$

$$\frac{dU_B}{dt} = -k_A \cdot U_A^2.$$

In this case, the quadratic model is more suitable, as it accounts for the nonlinear impact (e.g., scalability of strategies).

Adaptation of the Lancaster Model to the Problem

The Lancaster model allows for formalizing the competition between LLM_A and LLM_B through differential equations, taking into account the nonlinear influence of users and energy. This enables a deeper analysis of system interaction dynamics.

Answer: The Lancaster model for competition between LLM_A and LLM_B is formulated using differential equations that consider the nonlinear impact of users and energy.

1. System Metrics

- $U_A(t)$ – number of users of LLM_A at time t .
- $U_B(t)$ – number of users of LLM_B at time t .
- $E_A(t)$ – energy of LLM_A at time t .
- $E_B(t)$ – energy of LLM_B at time t .

Assume the total number of users is constant:

$$U_A(t) + U_B(t) = 1.$$

The Lancaster model for competition between LLM_A and LLM_B can be written as:

$$\frac{dU_A}{dt} = -k_B \cdot U_B^2 + k_A \cdot U_A^2,$$

$$\frac{dU_B}{dt} = -k_A \cdot U_A^2 + k_B \cdot U_B^2.$$

where: k_A, k_B – strategy effectiveness coefficients of LLM_A and LLM_B , U_A^2 and U_B^2 representing the nonlinear impact of user count on competition outcomes.

To account for energy (E_A, E_B), we introduce a modification:

$$\frac{dU_A}{dt} = -k_B \cdot U_B^2 \cdot E_B + k_A \cdot U_A^2 \cdot E_A,$$

$$\frac{dU_B}{dt} = -k_A \cdot U_A^2 \cdot E_A + k_B \cdot U_B^2 \cdot E_B.$$

This reflects the fact that strategy effectiveness depends on available energy.

Example of Initial Conditions:

$U_A(0) = 0.5, U_B(0) = 0.5$ (initial equilibrium).

$E_A(0) = 0.5, E_B(0) = 0.5$ (initial energy levels).

A system with more users (U_A or U_B) gains an advantage because its influence grows nonlinearly (U^2).

Energy (E_A, E_B) enhances strategic effectiveness, allowing the system with more energy to gain an advantage more quickly.

Eventually, one of the systems (LLM_A or LLM_B) may achieve complete dominance ($U_A = 1, U_B = 0$ or vice versa).

Calculation Using the Lancaster Model

A quadratic version of the Lancaster model is described, which is better suited for modeling nonlinear interactions between systems.

Initial Data

Initial number of users:

$$U_A(0) = 0.5,$$

$$U_B(0) = 0.5.$$

Initial energy:

$$E_A(0) = 0.5,$$

$$E_B(0) = 0.5.$$

Strategy effectiveness coefficients:

$$k_A = 0.2 \text{ (effectiveness of } LLM_A)$$

$$k_B = 0.15 \text{ (effectiveness of } LLM_B)$$

Total numbers of iterations:

$$N = 10.$$

Differential equations:

$$\frac{dU_A}{dt} = -k_B \cdot U_B^2 \cdot E_B + k_A \cdot U_A^2 \cdot E_A,$$

$$\frac{dU_B}{dt} = -k_A \cdot U_A^2 \cdot E_A + k_B \cdot U_B^2 \cdot E_B.$$

Constraints:

$U_A(t) + U_B(t) = 1$ (the total number of users is constant).

Iterative Calculation

Iteration 1:

Calculation of user change:

For U_A :

$$\frac{dU_A}{dt} = -k_B \cdot U_B^2 \cdot E_B + k_A \cdot U_A^2 \cdot E_A,$$

$$\frac{dU_A}{dt} = -(0.15) \cdot (0.5)^2 \cdot (0.5) + (0.2) \cdot (0.5)^2 \cdot (0.5) = 0.00625.$$

For U_B :

$$\frac{dU_B}{dt} = -k_A \cdot U_A^2 \cdot E_A + k_B \cdot U_B^2 \cdot E_B,$$

$$\frac{dU_B}{dt} = -(0.2) \cdot (0.5)^2 \cdot (0.5) + (0.15) \cdot (0.5)^2 \cdot (0.5) = -0.00625.$$

Value update:

$$U_A = U_A(0) + \Delta U_A = 0.5 + 0.00625 = 0.50625,$$

$$U_B = U_B(0) + \Delta U_B = 0.5 - 0.00625 = 0.49375.$$

Iteration 2:

Calculation of user change:

For U_A :

$$\frac{dU_A}{dt} = -(0.15) \cdot (0.49375)^2 \cdot (0.5) + (0.2) \cdot (0.50625)^2 \cdot (0.5) = 0.00732.$$

For U_B :

$$\frac{dU_B}{dt} = -(0.2) \cdot (0.50625)^2 \cdot (0.5) + (0.15) \cdot (0.49375)^2 \cdot (0.5) = -0.00732.$$

Value update:

$$U_A = 0.50625 + 0.00732 = 0.51357,$$

$$U_B = 0.49375 - 0.00732 = 0.48643.$$

Iteration 3:

Calculation of user change:

For U_A :

$$\frac{dU_A}{dt} = -(0.15) \cdot (0.48643)^2 \cdot (0.5) +$$

$$+(0.2) \cdot (0.51357)^2 \cdot (0.5) = 0.00850.$$

For U_B :

$$\frac{dU_B}{dt} = -(0.2) \cdot (0.51357)^2 \cdot (0.5) +$$

$$+(0.15) \cdot (0.48643)^2 \cdot (0.5) = -0.00850.$$

Value update:

$$U_A = 0.51357 + 0.00850 = 0.52207,$$

$$U_B = 0.48643 - 0.00850 = 0.47793.$$

Continuation of iterations up to iteration 10:

We repeat similar calculations for each subsequent step. Here are the results:

Iterations	U_A	U_B
1	0.50625	0.49375
2	0.51357	0.48643
3	0.52207	0.47793
4	0.53180	0.46820
5	0.54287	0.45713
6	0.55548	0.44452
7	0.56992	0.43008
8	0.58654	0.41346
9	0.60585	0.39415
10	0.62860	0.37140

After 10 iterations:

$$U_A = 0.6286 \text{ (62.86\% users).}$$

$$U_B = 0.3714 \text{ (37.14\% users).}$$

According to the Lancaster model, LLM_A gains an advantage due to its higher

effectiveness coefficient ($k_A > k_B$) and the nonlinear impact of user count. LLM_B gradually loses users, although the difference was initially small.

Answer: After 10 iterations, LLM_A has 62.86% of the users, while LLM_B has 37.14%.

Modeling Boyd Cycles

Boyd cycles (OODA-loop: Observe-Orient-Decide-Act) can also be used to model competition between two systems (LLM_A and LLM_B). This approach accounts not only for numerical metrics but also for strategic thinking and adaptability of each system. Let us consider how this can be formalized mathematically.

Main Stages of the Boyd Cycle

The Boyd cycle consists of four stages:

Observe: Collection of information about the environment (e.g., number of users, energy levels, satisfaction, etc.).

Orient: Analysis of collected data and determination of strategies based on the current context.

Decide: Selection of the best strategy to achieve goals.

Act: Implementation of the selected strategy.

This cycle is repeated iteratively, allowing systems to quickly adapt to changes in the environment.

System entities:

– $U_A(t)$, $U_B(t)$ – number of users of LLM_A and LLM_B at time t .

– $E_A(t)$, $E_B(t)$ – energy of LLM_A and LLM_B at time t .

– $S_A(t)$, $S_B(t)$ – user satisfaction of LLM_A and LLM_B at time t .

Boyd cycles allow for formalizing the competition between LLM_A and LLM_B through an iterative process of observation, analysis, decision-making, and action. This approach accounts for the strategic behavior of systems and their adaptability to changes in the environment.

Observe:

This phase involves gathering information about the state of the environment. For each system (LLM_A and LLM_B), we can formalize this process through a function:

$$Input_A = F_{observe}(U_A(t), E_A(t), S_A(t)),$$

$$Input_B = F_{observe}(U_B(t), E_B(t), S_B(t)),$$

where:

- $U_A(t), U_B(t)$ – number of users at time t .
- $E_A(t), E_B(t)$ – system energy at time t .
- $S_A(t), S_B(t)$ – user satisfaction at time t .
- $F_{observe}$ – a function that collects data about the system state.

Orient:

At this stage, the system analyzes the collected data and generates strategies. Using the "Function" primitive, we can write:

$$Strategy_A = F_{orient}(Input_A),$$

$$Strategy_B = F_{orient}(Input_B),$$

Where F_{orient} is a function that analyzes input data and generates a strategy.

For example, a strategy can be defined using a conditional operator ("Condition"):

$$\begin{aligned} Strategy_A &= \\ &= \begin{cases} \text{"Improve customization", if } U_A(t) > U_B(t); \\ \text{"Improve speed", if } U_A(t) \leq U_B(t). \end{cases} \end{aligned}$$

Similarly for $Strategy_B$.

Decide:

At this stage, the system selects the best strategy to achieve its goals. Using the "Condition" primitive, one can write:

$$\begin{aligned} Decision_A &= \\ &= \begin{cases} \text{"Improve customization", if } S_A(t) > S_B(t); \\ \text{"Improve speed", if } S_A(t) \leq S_B(t). \end{cases} \end{aligned}$$

Similarly for $Decision_B$.

Act:

At this stage, the system implements the selected strategy. The impact of strategies on users can be described through differential equations:

$$\begin{aligned} \frac{U_A(t)}{dt} &= k_A \cdot (Effect_{Decision_A}) - k_B \cdot (Effect_{Decision_B}); \\ \frac{U_B(t)}{dt} &= -\frac{U_A(t)}{dt}, \end{aligned}$$

where k_A, k_B – the efficiency coefficients of strategies LLM_A and LLM_B , $Effect_{Decision}$ – the efficiency of the selected strategy.

Boyd Cycle Composition

Primitive "Cycle"

The Boyd cycle can be implemented using the primitive "Cycle," described in your methodology file:

$$P(S) = \bigcup_{i=1}^N F(s_i),$$

where $S = \{Observe, Orient, Decide, Act\}$ – is the set of stages in the cycle, and $F(s_i)$ – is the operation performed at each stage.

Full System

Each system (LLM_A and LLM_B) performs its own Boyd cycle independently, but their actions interact through a shared environment:

$$\frac{dU_A}{dt} = f(U_A, U_B, E_A, E_B);$$

$$\frac{dU_B}{dt} = g(U_A, U_B, E_A, E_B).$$

Boyd cycles allow formalizing the competition between LLM_A and LLM_B through an iterative process of observation, analysis, decision-making, and action. This approach accounts for the strategic behavior of systems and their adaptability to changes in the environment.

Numerical Calculation Example

Let the initial conditions be given as:

$$\begin{aligned} U_A(0) &= 0.6, \quad U_B(0) = 0.4; \\ E_A(0) &= 0.55, \quad E_B(0) = 0.45; \\ S_A(0) &= 0.85, \quad S_B(0) = 0.75; \\ k_A &= 0.25, \quad k_B = 0.2. \end{aligned}$$

Iteration 1:

1. Observe:

$$Input_A = \{0.6, 0.55, 0.85\},$$

$$Input_B = \{0.4, 0.45, 0.75\}.$$

2. Orient:

$$Strategy_A = \text{"Improve customization"};$$

$$Strategy_B = \text{"Improve speed"}.$$

3. Decide:

$$Decision_A = \text{"Improve customization"};$$

$$Decision_B = \text{"Improve speed"}.$$

4. Act:

Efficiency of strategies:

$$Effect_{Decision_A} = 0.03,$$

$$Effect_{Decision_B} = 0.02.$$

Change in number of users:

$$\frac{U_A(t)}{dt} = 0.25 \cdot 0.03 - 0.2 \cdot 0.02 =$$

$$= 0.0075 - 0.004 = 0.0035,$$

$$\frac{U_B(t)}{dt} = -0.0035.$$

New values:

$$\frac{U_A(t)}{dt} = k_A \cdot (Effect_{Decision_A}) -$$

$$-k_B \cdot (Effect_{Decision_B}),$$

$$\frac{U_B(t)}{dt} = -\frac{U_A(t)}{dt},$$

$$U_A = 0.6 + 0.0035 = 0.6035,$$

$$U_B = 0.4 - 0.0035 = 0.3965.$$

Thus, Boyd's cycles allow to formalize the competition between LLM_A and LLM_B through an iterative process of observation, analysis, decision-making, and action. This approach takes into account the strategic behavior of systems and their adaptability to changes in the environment.

Comparative Analysis of LLM Competition Modeling Methods

For an objective evaluation of the effectiveness of the considered approaches to modeling competition between artificial intelligence systems, a set of criteria reflecting practical aspects of their application was defined. The main evaluation parameters included: implementation complexity, adaptability to environmental changes, ability to account for strategic behavior, predictive effectiveness, and universality of application.

The analysis showed that each of the considered methods has its own distinctive features and areas of optimal application (see Table 1). In particular, differential equations, despite their high implementation complexity, demonstrate the best predictive effectiveness due to their precise mathematical foundation. However, their adaptability to environmental changes is limited due to the static nature of the base equations.

The Lancaster model occupies an intermediate position between differential equations and Boyd cycles across most criteria. It allows for partial consideration of strategic behavior and shows satisfactory adaptability, although it is outperformed in this aspect by the Boyd cycles.

Table 1. Comparative Characteristics of Modeling Methods

Criterion	Diff. Equations	Lankaster Model	Boyd Cycles
Implementation Complexity	High	Medium	Low
Adaptability	Limited	Partial	High
Strategy Consideration	High	Partial	Full
Predictive Effectiveness	High	Medium	High
Universality	Limited	Partial	High

Special attention should be given to the advantages of Boyd cycles in the context of practical application. This method stands out due to its high adaptability and universality, thanks to the built-in mechanism of iterative strategy improvement. Despite relatively low implementation complexity, it provides high predictive effectiveness through a well-defined decision-making formalization.

The obtained results indicate that the choice of a specific method should depend on the objectives and available resources. In practice, combining different approaches is often the most effective way to maximize the efficiency of modeling competition between LLMs.

Conclusions

This article presents a comprehensive approach to modeling competition between AI systems, combining mathematical models, no-code tools, and an extended system of metrics. The study addresses the problem of simulating competition between two artificial intelligence systems, (LLM_A and LLM_B), which interact in a shared environment under limited resources such as users and energy.

The research tackles key tasks related to analyzing interaction dynamics, formalizing strategic behavior, and predicting competition outcomes. A comprehensive modeling approach is proposed using differential equations, the Lancaster model, and Boyd cycles (OODA-loop). A metric system has been developed that accounts for the number of users, satisfaction levels, response accuracy, processing speed, and energy efficiency.

To formalize strategic behavior, a no-code toolkit was employed to create prompts implementing the primitives "Condition," "Loop," and "Function." This enabled the description of complex system strategies in natural language terms. Based on Boyd cycles, it was demonstrated how adaptability and decision-making speed influence system success in competitive environments.

An extended metric system was introduced during simulation, incorporating not only user count and energy but also important behavioral aspects such as user satisfaction, response accuracy, query processing speed, and ethical considerations.

This work combines mathematical models (e.g., the Lancaster model) with no-code tools for the first time to formalize AI system strategies. This integration allows for analysis of both numerical and logical aspects of competition.

The results can be applied to predict AI system behavior in real-world scenarios such as information campaigns, cyber conflicts, and resource optimization in digital environments.

Future research directions include studying multi-party competition and incorporating additional factors such as social networks, cultural audience characteristics, and global trends influencing user decision-making.

In the future, these findings may be adapted to practical tasks such as combating disinformation, cybersecurity, and managing digital ecosystems.

The proposed models have potential practical applications in AI system management, helping developers predict system behavior, optimize strategies, and support efficient resource utilization.

The research was supported by a grant from the National Research Foundation of Ukraine (project registration number 2023.04/0087).

References

- [1] Haryanto, Christoforus Yoga, Anne Maria Elvira, Trung Duc Nguyen, Minh Hieu Vu, Yoshiano Hartanto, Emily Lomempow, and Arathi Arakala. "Contextualized AI for Cyber Defense: An Automated Survey using LLMs." In 2024 17th International Conference on Security of Information and Networks (SIN), pp. 1-8. IEEE, 2024. DOI: 10.1109/SIN63213.2024.10871242
- [2] Dmitry Lande, Yuriy Danyk. Competitive Artificial Intelligence in Information and Cyber Warfare Available at SSRN: <https://ssrn.com/abstract=5084698>, DOI: 10.2139/ssrn.5084698 (January 06, 2025). – 16 pp.
- [3] Yang, Yang, Bo Tian, Fei Yu, and Yuanhang He. "An Anomaly Detection Model Training Method Based on LLM Knowledge Distillation." In 2024 International Conference on Networking and Network Applications (NaNA), pp. 472-477. IEEE, 2024. DOI: 10.1109/NaNA63151.2024.00084
- [4] Grant Wilkins, Srinivasan Keshav, Richard Mortier. Hybrid Heterogeneous Clusters Can Lower the Energy Consumption of LLM Inference Workloads. e-Energy '24: Proceedings of the 15th ACM International Conference on Future and Sustainable Energy Systems. Pages 506 – 513. DOI: 10.1145/3632775.3662830
- [5] Dmitry Lande, Leonard Strashnoy Resilient Artificial Intelligence Architecture. ResearchGate: https://www.researchgate.net/publication/389497329_Resilient_Artificial_Intelligence_Architecture, DOI: 10.13140/RG.2.2.36062.55360 (Mart 02, 2025). – 9 pp.
- [6] Lancaster, Kelvin. "The dynamic inefficiency of capitalism." *Journal of Political Economy* 81.5 (1973): 1092-1109. DOI: 10.1086/260108
- [7] Mancillas, James, and Gregory L. Cantwell. "Integrating Artificial Intelligence into Military Operations: A Boyd Cycle Framework." Strategic Studies Institute, US Army War College (2017). URL: <http://www.jstor.com/stable/resrep12117.11>
- [8] Dmitry Lande, Leonard Strashnoy Semantic AI Framework for Prompt Engineering. ResearchGate: https://www.researchgate.net/publication/389692528_Semantic_AI_Framework_for_Prompt_Engineering. DOI: 10.13140/RG.2.2.21116.24964 (Mart 9, 2025). - 14 pp.