Towards Scalable Multi-Agent Simulations: Predicting LLM-Agent Responses

HAW HAMBURG

MARS GROUP
Multi-Agent Research & Simulation

Motivation

Large Language Model (LLM) agents can simulate realistic, human-like decision-making. This makes them a promising tool for exploring diverse urban scenarios and stakeholder behaviors. However, LLM agents are computationally expensive and do not scale well to large populations. This project investigates an alternative approach that approximates the behavior of LLM agents, enabling scalable multi-agent simulations of complex environments.

Environment

To simulate realistic urban dynamics, we use a simulation framework with a detailed, city-scale scenario. The MARS (Multi-Agent Research & Simulation) framework provides the general infrastructure for multi-agent simulations. It offers a spatially and temporally structured environment where agents can perceive, act, and interact. The Smart Open Hamburg (SOH) scenario, models the mobility behavior of urban agents in the city of Hamburg. The SOH scenario defines the city's geography, infrastructure, and activity patterns, enabling agents to exhibit realistic, human-like mobility.

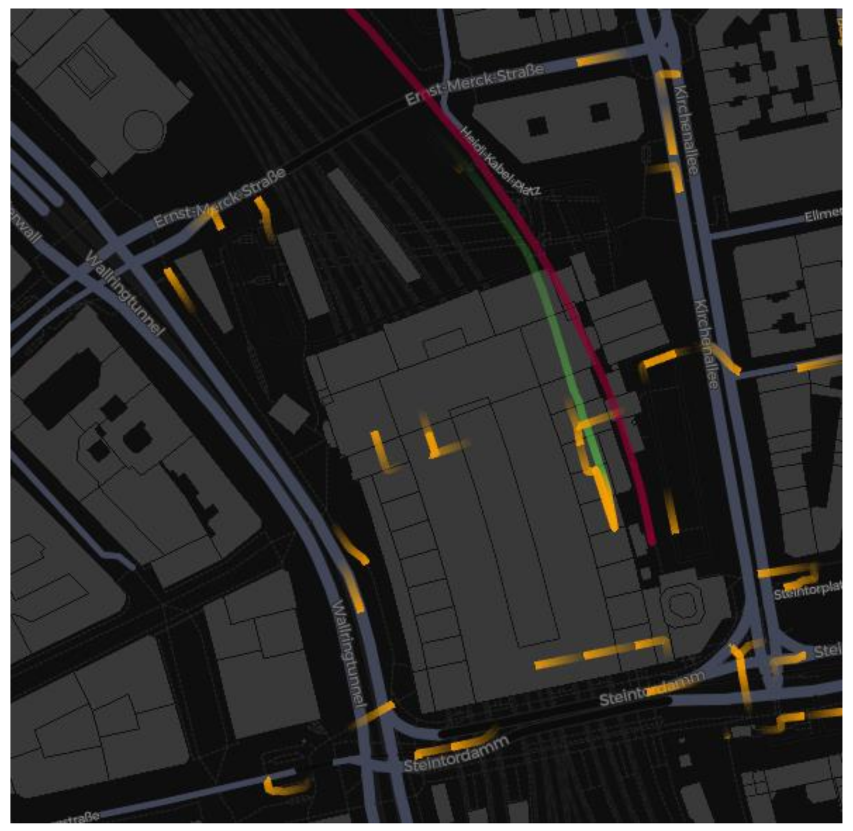


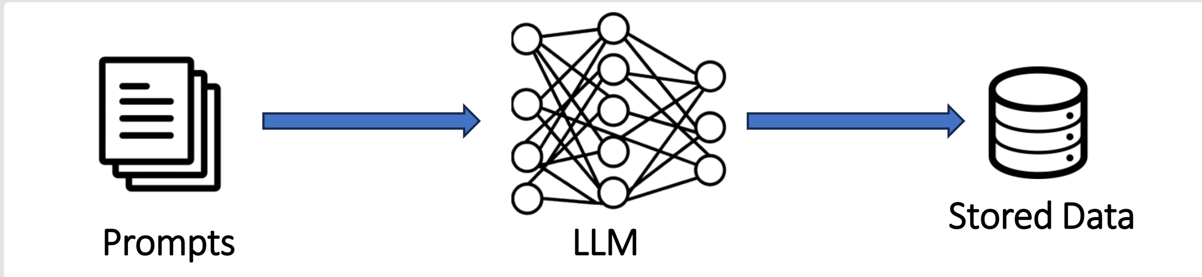
Figure 1. Simulation of Agents at Hamburg Central Station

Methods

To investigate whether the probability of an agent's decision can be predicted from individual traits and situational factors, an encoder-only transformer model will be developed. The model receives both the modular probabilities and the semantic representations of the selected traits and situations as input, and predicts the combined scenario probability as output.

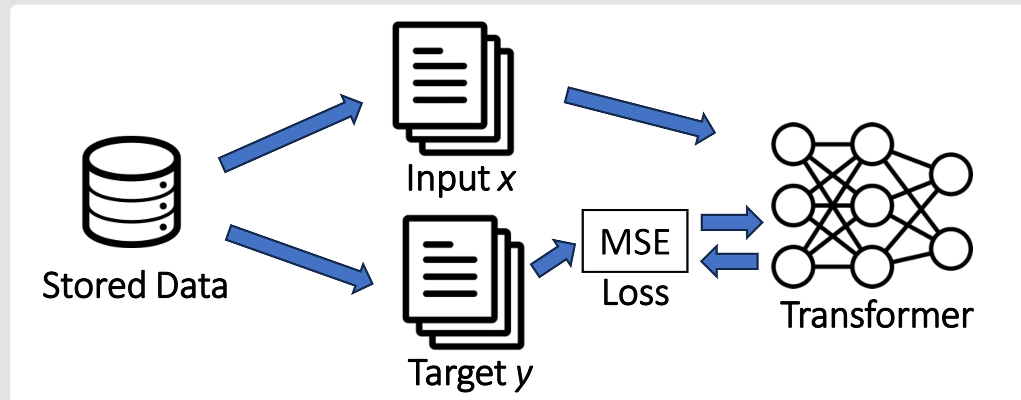
1. Data Generation

- Define situations ***S***, agent traits ***T*** and action ***a***
- Prompt structure (simplified): Given selected traits ***T'***, selected situations ***S'***, and an action ***a***, return the probability of ***a***
- Generate LLM responses for each individual trait or situation
- Generate LLM responses for the full scenarios
- Store all inputs, and corresponding LLM outputs as structured data



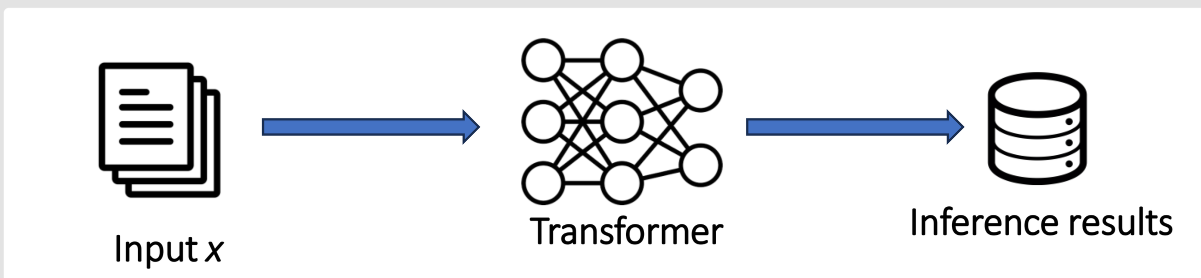
2. Training

- Train an encoder-only transformer on the structured dataset:
- Input ***x***: modular probabilities + embeddings of selected traits ***T'*** and situations ***S'***
- Target ***y***: LLM's combined scenario probability



3. Inference

- Use the trained encoder-only transformer to predict the combined scenario probability
- Input ***x***: modular probabilities + embeddings of new combinations of ***T'*** and situations ***S'***
- Store inference results



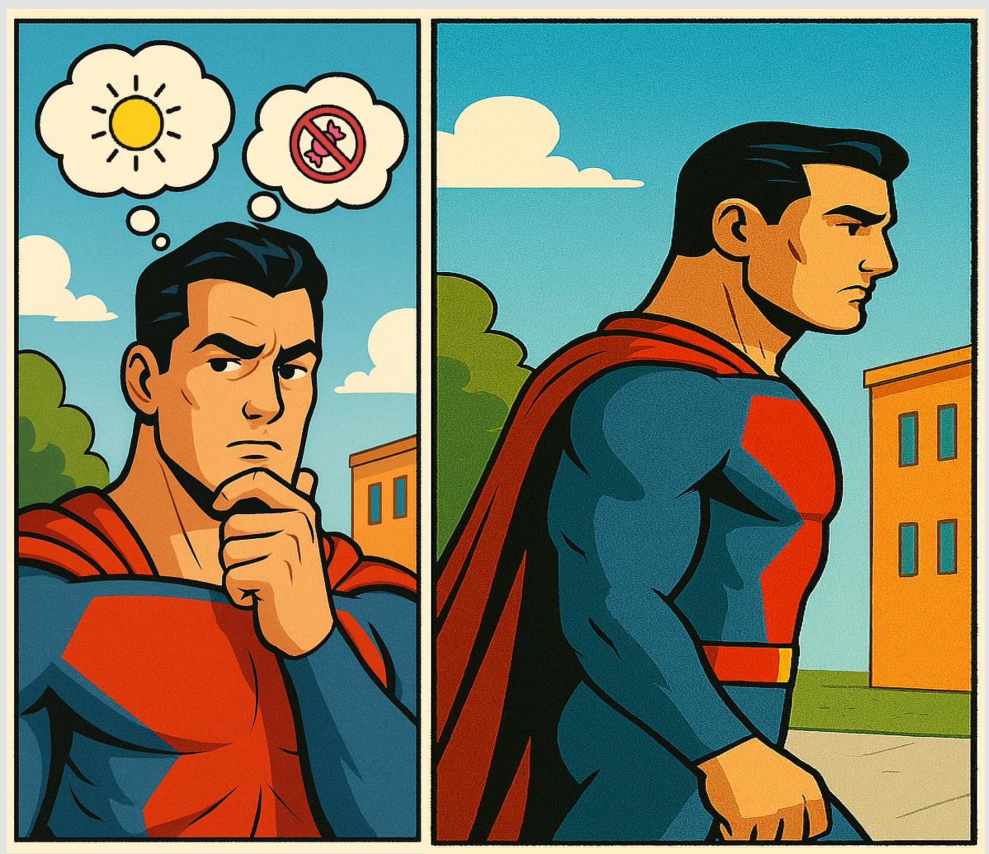
4. Architecture Optimization

- Compare inference results across different model configurations
- Adjust architecture parameters (e.g., number of layers, attention heads, hidden size) based on performance
- Repeat training and inference until the model accurately captures the interaction between ***T'***, ***S'***, and ***a***, and the best architecture is identified
- The best architecture minimizes the prediction error: $L = MSE(P_{Transformer}(a \mid T', S'), P_{LLM}(a \mid T', S'))$

Agent Decision Depends on Traits

Situation: ice cream truck nearby

Trait: likes treats→ Action: buys ice cream



Trait: dislikes treats→ Action: walks away

Case study: SOH

At the start of the simulation, the modular probabilities of each trait and situation relevant to the targeted behavior are queried from the LLM once and stored. Regardless of the number of agents in the simulation, no further expensive LLM queries are required. During the simulation, each agent's current state is fed into the trained transformer model, which efficiently predicts the agent's decision. This allows simulating heterogeneous agents, each with unique traits and contexts at a fraction of the computational cost of repeated LLM queries.

Related Work

Recent research has explored methods to make LLM-based multi-agent simulations more computationally efficient. [1] and [2] address the high computational cost of simulating large numbers of LLM agents by grouping agents into representative clusters and assigning group-level decisions in-stead of querying each agent individually. However, these approaches are validated only at the macro level. Moreover, their simplifications reduce individuality, making it difficult to capture interactions with the environment.

Next Steps

1. Integrate the model into the SOH simulation environment.
2. Investigate whether the model can predict alternative, similar actions by leveraging the similarity between actions in relation to traits and situations.

References

[1] Chopra A. et al. (2025). *On the Limits of Agency in Agent-based Models.* AAMAS 2025. DOI: 10.5555/3709347.3743565

[2] Yan et al. (2024). *OpenCity: A Scalable Platform to Simulate Urban Activities with Massive LLM Agents.* DOI: 10.48550/arXiv.2410.21286

Contact

Ersan Baran (M.Sc)
Ersan.Baran@haw-hamburg.de