1. Overview

- The Matrix is an agent-based modeling (ABM) framework
- The Matrix is free and open source software
- Can be used to model ‘hybrid’ time simulations
  - A combination of discrete time and discrete event
  - Specialized for ‘compute and data intensive’ simulations
- Successfully used to model 3M individual cognitive agents
  - A three order of magnitude increase over previous studies

2. Capabilities of the Matrix

- Allows writing agents in popular programming languages
  - Python, R, C, C++, Java, Lisp, …
- Supports use of GPU units, and popular neural network libraries
  - TensorFlow, PyTorch, Keras, Lens, …
- Enables use of cognitive system libraries like ACT-R
- Supports running simulations on cloud platforms
  - Amazon EC2, Google Compute Cloud, and Microsoft Azure
- Gracefully handles large (∼ hundreds of gigabytes) system state

3. Models Implemented with the Matrix

<table>
<thead>
<tr>
<th>Name</th>
<th>Model Type</th>
<th>Prog. Lang.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Freq-Stat</td>
<td>Frequentist statistical model</td>
<td>Python</td>
</tr>
<tr>
<td>Soc-Th</td>
<td>Social structure theory model</td>
<td>Python</td>
</tr>
<tr>
<td>CM-ANN</td>
<td>Artificial neural network model</td>
<td>C++</td>
</tr>
<tr>
<td>CM-Bayes</td>
<td>Bayesian cognitive theory model</td>
<td>R</td>
</tr>
<tr>
<td>CM-ACTR</td>
<td>ACT-R cognitive theory model</td>
<td>Common Lisp</td>
</tr>
</tbody>
</table>

4. Formal Model of a Matrix Simulation

- System state: \( x(0) \rightarrow x(1) \rightarrow \cdots \rightarrow x(n) \)
- Transition function: \( x(t + 1) = g_{\text{sim}}(x(t)) \)
- Compute updates and apply: \( g_{\text{sim}}(x(t)) = g_{\text{act}}(x(t)), g_{\text{act}}(x(t)) \)
- Compute updates for \( v_j; g_{\text{act}}(x(t)) = \bigcup_{j \in V} g_{\text{act}}(x(t)) \)

5. The Matrix Runtime System

- Agent Synchronization Phase
  - Agents perform initializing steps before the simulation begins.
- Update Computation Phase
  - Agents compute updates, the controller begins to share output and post to the state store.
- Data Sharing Phase
  - Agent computations end for current time step.
- Controller continues to share output and post to the state store.
- Controller Synchronization Phase
  - Controller continues to post to the state store as it waits for other controllers to finish.

6. States of a Controller Process

- Intentional Module (aPFC)
- Declarative Module (Temporal Hippocampus)
- Procedural and Motor Modules
  - Matching (Striatum)
  - Selection (Pallidum)
  - Execution (Thalamus)
- Visual Buffer (Parietal Cortex)
- Manual Buffer (Motor/Cerebellum)
- Visual Module (Occipital/other)
- Manual Module (Motor/Cerebellum)

7. Simulating GitHub

- Learning: Learn parameters from training event trace
- Simulation: Generate simulated event trace

8. Conclusion

- The Matrix facilitates rapid prototyping of ‘compute and data intensive’ agent models
- The Matrix allows flexibility in use of programming languages and libraries
- Matrix simulations can run on commodity clusters and cloud computing platforms