Fall 9-12-2024

Neurosymbolic Cognitive Methods for Enhancing Foundation Model-based Reasoning

Kaushik Roy
kaushikr@email.sc.edu

Siyu Wu
Penn State University, sfw5621@psu.edu

Alessandro Oltramari
Bosch Center for Artificial Intelligence, alessandro.oltramari@us.bosch.com

Publication Info

Preprint version *Preprint*, Fall 2024.

August 2024

Neurosymbolic Cognitive Methods for Enhancing Foundation Model-based Reasoning

Kaushik ROY [a,1], Siyu WU [b] and Alessandro OLTRAMARI [c]

[a] *Artificial Intelligence Institute South Carolina*
[b] *College of Information Sciences and Technology, the Pennsylvania State University*
[c] *Bosch Center for Artificial Intelligence*
ORCiD ID: Kaushik Roy https://orcid.org/0000-0001-6610-7845, Siyu Wu
https://orcid.org/0009-0007-5903-4909, Alessandro Oltramari
https://orcid.org/0000-0003-1559-4852

Abstract. Foundation models have emerged as powerful tools, exhibiting extraordinary performance across various tasks, such as language processing, visual recognition, code generation, and human-centered engagement. However, recent studies have highlighted their limitations when grounded, abstract, and generalized reasoning capabilities are required. Complex tasks often involve multiple hierarchical reasoning steps, which are typical features of human thinking processes. In fact, in this chapter we claim that cognitively-inspired computational models, such as the so-called Common Model of Cognition, are key to enable complex reasoning within foundation model-based artificial intelligence (AI) systems. We investigate neurosymbolic approaches for mapping AI system components to those of the Common Model of Cognition, either fully or partially. Specifically, two pathways are explored: (i) Given a task and its solution, we explore the effect of fine-tuning foundation models on the output traces obtained through a cognitive architecture such as ACT-R. The hypothesis is that, after fine-tuning, the foundation model will more closely emulate the cognitive reasoning processes necessary to solve the specific task. (ii) In the second approach, given a task, we explore mapping the solution requirements to various components of the common model of cognition and invoke a combination of foundation model-based pattern recognition, external knowledge augmentation and control flow planning to facilitate cognitive reasoning for the task. The chapter covers the background of foundation models and the common model of cognition, a survey of the existing landscape in integrating foundation models and cognitive architectures, and a discussion of insights from preliminary implementations of the two neurosymbolic pathways across real-world and synthetic tasks.

Keywords. Foundation Models, Common Model of Cognition, Neurosymbolic AI

[1] Corresponding Author: Kaushik Roy, kaushikr@email.sc.edu

1. An Introduction to Foundation Models

Foundation models are large-scale machine learning models, typically based on deep learning architectures, that are trained on vast amounts of data and designed to be adaptable to a wide range of downstream tasks. These models are termed "foundation models" because they serve as a basis from which more specialized models can be derived or fine-tuned for specific applications [1].

Perception Key features of foundation models include (i) Scale: They are often extremely large, with billions or even trillions of parameters, enabling them to capture complex patterns and knowledge from the training data. (ii) Pre-training and Fine-tuning: Foundation models are usually pre-trained on a large and diverse dataset in a self-supervised or unsupervised manner. After pre-training, they can be fine-tuned on smaller, domain-specific datasets to perform particular tasks, such as natural language understanding, image recognition, etc. (iii) Versatility: Due to their broad training, foundation models can be adapted to perform a wide range of tasks, often with little modification. For example, a foundation model trained on general text data might be fine-tuned for tasks like sentiment analysis, translation, or question answering [2]. Foundation models span multiple modalities, for example, GPT (Generative Pre-trained Transformer) text models like GPT-3 and vision models like CLIP (Contrastive Language-Image Pre-training) [3]. Foundation Models represent a significant and disruptive advancement in AI, providing a great starting point for a wide range of AI applications such as natural language and vision processing tasks. We refer to such tasks as *perception* tasks [4].

Reasoning Despite their capabilities, foundation models can struggle with tasks involving complex reasoning, often requiring the ability to "understand" abstract concepts, draw connections between disparate pieces of information, and engage in multi-step, hierarchical "thinking" or "cognition", which we refer to as cognitive reasoning. Recent work has exposed the lack of structured and logical reasoning capabilities in foundation models [5,6]. For example, solving a complex mathematical problem often requires multiple reasoning steps, each building on the previous one. Foundation models have been shown to struggle with basic arithmetic and simple algebraic operations [7]. Recent progress towards such reasoning has been demonstrated by augmenting foundation models with external procedures such as policy search to enable multi-step problems, especially those involving abstract concepts like proofs or applying theorems (e.g., Deepmind's AlphaGeometry and AlphaProof) [8].

Fundamentally, the limitations of foundation models lie in their inherent design. These systems are statistical learners, optimized for pattern recognition rather than logical deduction or abstract thought [9]. While they can mimic certain aspects of reasoning through pattern-based learning, they often lack the deeper, structured understanding required for cognitive reasoning. Figure 1 illustrates a scenario where the GPT-3.5 model engages in problematic reasoning.

Grounding and Verifiability As mentioned, foundation models are statistical learners that rely on patterns in the data they are trained on. However, they don't inherently include any mechanism to verify whether those patterns are grounded in real-world facts or reflect human-like commonsense and reasoning. As a result, they can produce logically flawed or nonsensical outputs [11]. This behavior manifests in many concerning ways ranging from seemingly innocuous adverse outcomes such as *hallucinations* – generat-

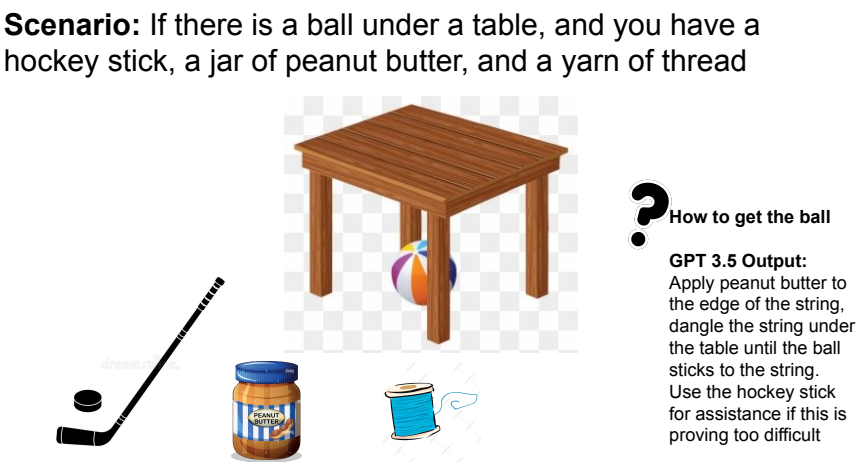

Figure 1. An example of problematic cognitive reasoning by GPT-3.5. The correct reasoning would result in simply poking the ball from under the table using the hockey stick, however, GPT-3.5 insists on using all three objects in a strange way to retrieve the ball from under the table (Example inspired-ny from Joshua B. Tenenbaum's AAAI 2023 Keynote Address)[10].

ing information that appears coherent and plausible but is entirely fabricated or incorrect, to more harmful outcomes such as *regurgitating harmful biases* – generating biased or discriminatory content, reflecting stereotypes and prejudices present in the training data [12]. For instance, they might associate certain professions with a particular gender or make assumptions based on race [13]. Fundamentally, foundation models generate content based on probabilities derived from training data, which can result in the confident assertion of falsehoods.

2. An Introduction to The Common Model of Cognition

The Common Model of Cognition (CMC) is a theoretical framework that presents a model of human cognition codified as a computational architecture [14]. The CMC is a brain-inspired framework validated by large-scale neuroscience data [15]: it identifies core components and processes fundamental to human cognition, including memory, perception, motor actions, and decision-making. The model assumes a cyclical process where these components interact to produce intelligent behavior.

Key components of the* CMC*:

1. **Perception:** The process of acquiring and transforming raw inputs from the environment into a representation useful for solving a given task.
2. **Working Memory:** A temporary storage system where task-relevant information is actively held and manipulated.
3. **Long-term Memory:** A more permanent storage system where factual syntactic and semantic knowledge is maintained.

4. **Procedural Memory:** Stores the necessary skills and procedures, formatted as rules (e.g., if-then rules) that can be applied for solving a given task.
5. **Motor or Action:** Executes solutions to the given task by applying the relevant rules from the procedural memory to the contents of the working memory.

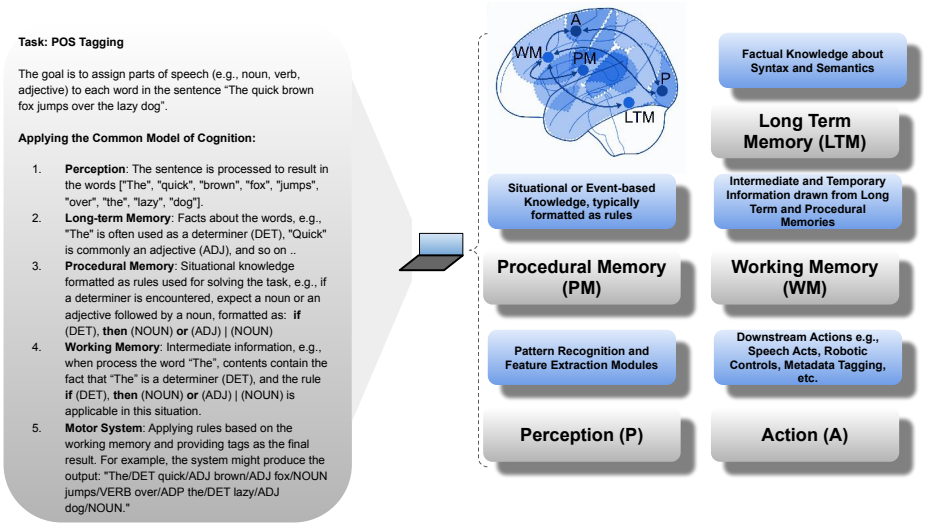

Figure 2. The Common Model of Cognition with an Illustration of Applying it to Solve Part-of-Speech Tagging on the Example Sentence "The quick brown fox jumps over the lazy dog".

The CMC provides a cognitively-grounded structured approach to problem-solving by breaking down the reasoning process into perceiving input, storing and manipulating information in working memory, reasoning based on available knowledge applied to the contents of the working memory, and manifesting responses by means of motor actions or speech acts. CMC-based reasoning processes aim to mirror how humans think about and understand the physical and social worlds, making it an attractive framework for building cognitively-inspired artificial intelligent systems. Figure 8 illustrates the CMC and how it maps to the human brain. The figure also provides an example of how the CMC is applied to solve a part-of-speech tagging task.

3. The Current Landscape of Methods for Integrating Foundation Models and Cognitively-Inspired Systems in AI Systems

The importance of integrating cognitively-inspired mechanisms into large scale machine learning models has been recently acknowledged by one of the key figures in deep learning, Yann LeCun: in a position paper published in 2022 [16], he described a biologically-inspired cognitive architecture, where a so-called *configurator* orchestrates information provided by different modules, such as the *perception module* and the *world model module*, which replicate the functions emerging from prefrontal-cortical processes. Furthermore, a *motivation model* – designed to mimic the role of the amygdala in producing basic emotional states like pain and pleasure – is used to compute intrinsic costs asso-

ciated with current and future actions, a mechanism that is instrumental to inform predictive capabilities. In line with the current trend of investigating computational models of cognition in the context of large-scale neural networks, a recent blog [17] provides an overview of how large language models (LLMs) could be used to control autonomous agents. **In general, the mutual benefits of integrating foundation models and cognitively-inspired systems are clear: on one hand, the former can provide the latter with the necessary scaffolding to solve tasks at scale, a well-known limitation of cognitive systems, which are heavily dependent on manual design and programming; on the other hand, the latter - especially through the CMC framework - can endow the former with a human-inspired computational framework for handling tasks requiring cognitive reasoning. Thus, it follows that a synergistic integration between the two approaches is key to support robust, highly flexible (or generalist) and scalable task reasoning within AI systems.**

Considerable effort has been devoted to bridging the processes typically associated with neural network-based representations, such as activations and vectors. Newell introduced a framework that divides a computational system into "bands" and "system-levels," corresponding to neural representations and traditional cognitive representations, such as chunks, symbols, high-level algorithmic descriptions, and heuristics [18]. Chunks, in this context, denote symbolic units of information communicated within a cognitive architecture, typically using limited-capacity buffers [19]. Newell's distinction is based on the premise that neural representations are learned from data through prediction-focused objectives, thus representing *stochastic* information. In contrast, cognitive representations are typically *declared* as production rules (which can be likened to if-else rules) and propositional statements (chunks), representing deterministic information. The fundamental difference between these representations—*stochastic* versus *declarative*—presents a significant challenge in integrating neural network-based and cognitive reasoning-based approaches [20].

Sumers et al. adopt a contemporary perspective based on LLMs, utilizing cognitive architecture-inspired task pipelines with *language agents*. This approach involves multiple language models, referred to as language agents, executing tasks through operations guided by language model-driven control logic (e.g., vector similarity-based external knowledge retrieval, function calling, or prompt-chaining) [21]. However, since contemporary LLMs fundamentally operate through predictive mechanisms, this approach assumes implicitly that all cognition is achieved through an auto-regressive predictive process. It has been demonstrated categorically that cognitive reasoning cannot be attained solely through predictive mechanisms [22]. Kelly and Reitter explore an integrated approach to bridging cognitive and neural representations using holographic declarative memories, wherein the declarative memory (long-term memory) module of the CMC is replaced with a distributional semantics model, such as an autoregressive language model. They argue that tokenization, encoding, and decoding operations provide effective translations between cognitive declarative information and neural stochastic information [23].

While we recognize the promise of the approaches discussed so far, as evidenced by experimental data, we propose that an *abstraction* layer should serve as an intermediate representation between neural and cognitive representations, rather than relying on direct translation between the two [24]. Recent work supports this notion, showing the emergence of "meta-optimizers" within LLMs and analyzing how these models operate when

provided with instructions [25]. Simply put, a LLM executes instructions in a manner akin to executing learned "metacognitive" rules, which follow a globally-contextualized meta-optimization step, as opposed to a locally-contextualized optimization step conveyed by gradients. More precisely, this type of instruction-following procedure operates through an intermediate *abstraction* layer that is distinct from predictions based on gradient information in neural representation spaces and from cognitive reasoning based on the symbolic information contained in cognitive representations. For example, the part-of-speech tagging task illustrated in Figure 8 provides an example of task-specific metacognitive rules-of-thumb.

Hypothesis Statement

In this chapter, we hypothesize that cognitive reasoning can be achieved using mechanisms similar to executing metacognitive rules-of-thumb, similar to instruction prompts in modern LLMs. This approach would allow us to leverage foundation models within the CMC *framework to effectively bridge neural and cognitive representations, thereby enabling cognitive reasoning in AI systems.*

In the following sections, we refine these ideas and propose a neurosymbolic framework for integrating foundation models with the CMC. Specifically, we suggest leveraging neurosymbolic mechanisms to implement semantic reasoning processes using metacognitive instruction-based reasoning, thereby facilitating the integration of foundation models (e.g., prompts) and the CMC (e.g., procedural memory) for enabling cognitive reasoning [26,27].

4. CMC-scaffolded Neurosymbolic AI for Cognitive Reasoning

Neurosymbolic AI has been conceptualized in various ways, with detailed categorizations of available implementation methods [28]. To ensure clarity, this work adopts a categorization similar to that proposed by Sheth et al., which includes: **(i) Compressing cognitive representations for integration with neural representations**, followed by neural pattern recognition; and **(ii) Mapping neural representations to cognitive representations, followed by instruction-based reasoning** [29]. The subsequent sections analyze specific implementations of these approaches within the CMC framework, applied to particular tasks, and evaluate them in relation to the **hypothesis statement** introduced in Section 3.

4.1. Compressing cognitive representations for integration with neural representations

In this experiment, we utilize the LLAMA architecture for neural representations and the ACT-R cognitive architecture for a system based on the CMC. This section first addresses the preliminary concepts, followed by a detailed description of our experiment and the conclusions related to the hypothesis, as presented in Section 3.

4.1.1. The ACT-R *Cognitive Architecture in Decision making*

The CMC integrates essential features from various cognitive architectures, which are computational frameworks designed to capture the invariant mechanisms of human cognition (for an introduction on cognitive architectures, see [30]). These mechanisms include functions related to attention, control, learning, memory, adaptivity, perception, and action. Cognitive architectures propose a set of fixed mechanisms to model human behavior, functioning akin to agents and aiming for a unified representation of the mind. By utilizing task-specific knowledge, these architectures not only simulate but also explain behavior through direct examination and real-time reasoning tracing. One representative cognitive architectures is ACT-R [31].

ACT-R encompasses perception, memory, goal-setting, and action. It uses two primary types of knowledge representations: declarative and procedural. Declarative knowledge comprises chunks of information, stored in declarative memory. Procedural knowledge, on the other hand, involves performing basic operations, moving data among buffers, and executing instructions. Over the years, ACT-R has accounted for a broad range of tasks at a high level of fidelity, reproducing aspects of complex human behavior, from everyday activities like event planning [32] and car driving [33], to highly technical tasks such as piloting an airplane [34], and monitoring a network to prevent cyberattacks [35]. The modeling approaches used include: *strategy or rule-based*, where different problem-solving strategies are implemented through various production rules, and successful strategies emerge on the basis of suitable reward functions [36,37]; e*xemplar or instance-based*, an approach that relies on past experiences stored in declarative memory to solve problems [38]; *hybrid*, which combine rule-based and instance-based approaches [39]. ACT-R was chosen for this study to provide the intermediate representations of cognitive reasoning steps. Three key features distinguish the use of ACT-R in creating models for decision-making tasks that involve learning:

- *Self-configuration*: ACT-R efficiently translates instructions into structured rules, forming the basis for task-specific production rules that enhance the efficiency of task execution.
- *Modular design mirroring human cognition*: ACT-R's modules emulate human cognitive functions - perceptual modules update the system's view of the environment, a goal module tracks progress towards objectives, a declarative module uses past experiences for contextual understanding, and a central buffer system enables communication between modules. Additionally, the central production system recognizes patterns to initiate coordinated actions.
- *Subsymbolic processes for decision-making*: ACT-R excels in its ability to reliably retrieve relevant memories and activate appropriate rules, ensuring both efficient and adaptive performance in decision-making tasks, such as skills training. It does so at a pace that mirrors human performance and offers the opportunity to model learning during this process.

4.1.2. Problem Definition: Design for Manufacturing

We define the terminology that constitutes our problem. The problem setting is a prototypical manufacturing production-line workflow, from supplier to customer, for which there exists a Value Stream Map (VSM; see Figure 3), which allows for tracking the efficiency at different sectors of the process and abstracts the overall problem for mathemat-

ical modeling and optimization. Key sectors include: Body Production, Pre-Assembly, Assembly, Honing, Washing, Testing, and Packaging. Early sectors pose potential efficiency problems in the workflow and may warrant optimization (triangles), while later stages are governed by *First-In-First-Out* (FIFO) processes. The metrics at each stage include Cycle Time (CT), Overall Equipment Effectiveness (OEE), and Mean Absolute Error (MAE); the flow progresses through each stage, aiming for efficient operation, performance monitoring, and error minimization to ensure high-quality production output and timely customer delivery.

Focused on maintaining stable output for manufacturing plants, we consider plant managers' feedback alongside the VSM structure to define two decision-making problems that aim to reduce Total Assembly Time (TAT) while minimizing Total Defect Rate (TDR). An agent $\mathscr{G}$ is a predictive model that takes a natural language question $\mathscr{Q}$ as a prompt, along with N snapshots of the sector-wise production flow data $\{$CT, OEE, MAE$\}$. In a *single-facet decision-making problem*, $\mathscr{G}$ outputs a binary decision (0 or 1) on which of two sectors, pre-assembly or assembly, requires a time reduction. In a more-challenging *multi-faceted decision-making problem*, $\mathscr{G}$ should output the same binary decision as before, about which sector should be the optimization target, along with an optimization *strategy* S. Here, S is a strategy defined by one of several decision-making personas that govern manufacturing process management, which we refer to in the manuscript as 'novice', 'intermediate', and 'expert'.

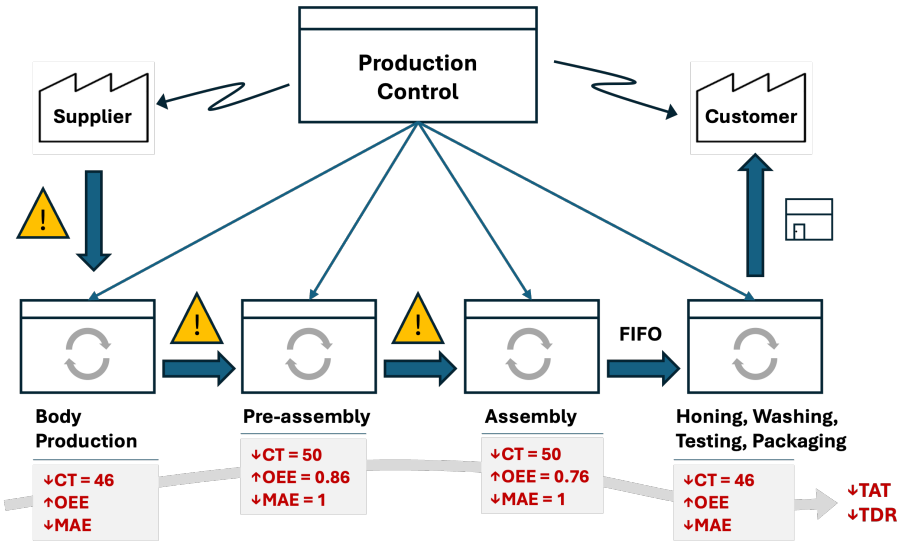

Figure 3. A *Value Stream Map* of manufacturing process.

4.1.3. A Cognitive Model For the Manufacturing Domain

In recent work, we released VSM-ACTR 2.0 (hereafter referred to as VSM-ACTR), a rule-based ACT-R cognitive model for decision support in manufacturing. VSM-ACTR has incorporated the meta-cognitive processes that reflect on and evaluate the progress of chosen strategies—with an emphasis on headcount cost evaluation, through a reward

structure that enables a process akin to reinforcement learning. This system allows the model to dynamically assess the impact of headcount costs on decision-making outcomes, computing a reward or penalty for each decision cycle. These rewards or penalties then propagate back to the initial production rule that initiated the decision cycle, thereby dynamically adjusting the utility of each decision-making strategy. VSM-ACTR integrates the prototypical decision process with insights into how cognitive models represent different levels of expertise [40,41], categorizing users into three levels of expertise: novices, intermediates, and experts. Novices engage in decision-making using intuitive deliberative chunks. Intermediates can manage key metrics such as CT and OEE but struggle with the systematic analysis of intertwined variables. Experts, on the other hand, make judgments systematically. The cognitive model employs three types of knowledge chunks: decisions, decision merits, and goals. The 'decision chunk' encodes eight slots including reduction time (goal), decision-making state (novice, intermediate, expert), and related variables. The 'decision merits chunk' holds information on sector weights, defect increases by sector, and comparative defect rate increases. The 'goal chunk' captures the initial production conditions and the ultimate goal of achieving the optimal decision. In addition, the model uses 18 procedural rules driven by goal-focused objectives across 20 states, covering actions such as choosing strategies, actions, working memory management, decisions, and evaluations.

Production Rule Sets Three sets of production rules represent the decision-making behaviors of novice, intermediate, and expert decision-makers. We use the expert production rule set as an example, once the decision-choice center decides to activate a set of expert decision productions, the process begins by perceiving the problem and retrieving related decision-making metrics from chunks. The imaginal buffer then acts as a working memory platform, holding and manipulating relevant information during the decision-making process. It allows the model to construct new mental representations or modify existing ones based on incoming data or problem-solving needs. This involves using the imaginal buffer to assess the relationships between the decision target and decision metrics, particularly considering the impact of each sector's weight on the defect rate change, and determining the final defect rate increase for each sector. These results are stored in the imaginal buffer and later retrieved for comparison. This enables the model to select the sector with the lowest defect increase. After one decision-making cycle, the model evaluates the headcount cost, rewarding or penalizing the entire process based on the evaluation results and decision strategy used before looping back to the next decision-making round.

Level of Expertise Mechanism The model can learn while performing tasks through two mechanisms leading to varying levels of expertise through differentiating knowledge representations, as shown in figure 4. **Declarative Memories**: These memories store knowledge that aligns with human intuition and expertise gained from the VSM. For example, the green triangles in the figure represents a portion of the intuition used by novice decision-makers. **Production Rules**: These rules capture the rational decision-making processes observed in human subjects. The green lines illustrate how the imaginal buffer retrieves relevant portions of the novice declarative memory and feeds them to the novice production rule set. Intermediate and expert decision-making levels follow the same principle. Red and blue shapes represent their respective declarative memory chunks, and the corresponding colored arrows show the flow of information through

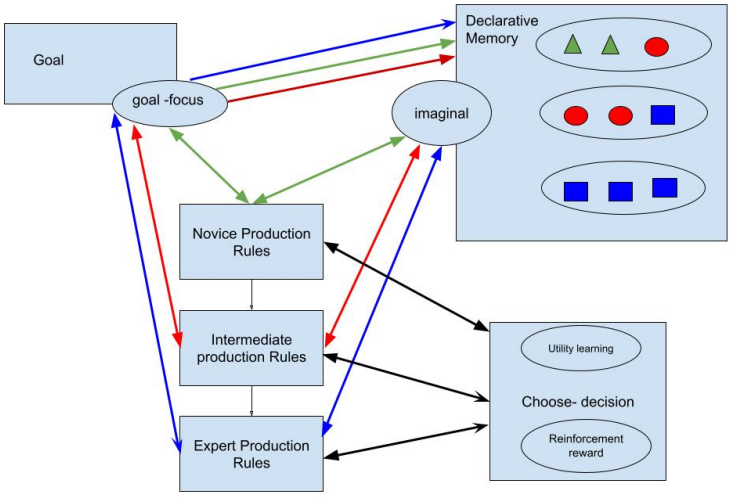

Figure 4. Level of expertise mechanism in VSM-ACT-R

their production rule sets. Finally, the goal buffer utilizes the "goal focus" command to manipulate the different phases of the task.

Beyond mimicking human behavior, the model also simulates the learning progress achieved by the **Decision-Choice Control**, which manages errors, learning, and memory through utility learning and reinforcement rewards. Novice decision-making starts with a utility base and includes a noise setting. The intermediate and expert production rules receive rewards when the corresponding decision-making results are achieved. The utility of these production rules updates is based on the rewards received and the retention of memory, which depends on the time passed since the rule last fired.

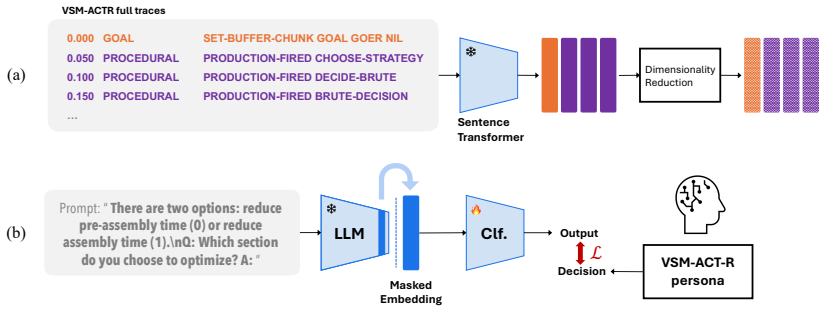

Figure 5. (a) Obtaining decision representations from VSM-ACT-R. (b) LLM feature extraction for behavior prediction.

VSM-ACTR model evaluation We ran the VSM-ACTR model across 2012 decision-making trials and 32 problem sets to analyze its behavior [42]. Each model run comprised 15-16 trials until reach a more stable expert behavior. We encoded decision types as 0, 1, and 2 for novice, intermediate, and expert strategies, respectively.

To assess learning, individual differences, and progression, we initially used descriptive statistics to chart the average progression of decision types over 16 trials. We then

employed a mixed linear model to evaluate the influence of trial numbers on decision types, incorporating repeated measures and random effects to account for individual variance. Additionally, an ordered logistic regression analyzed the relationship between the number of trials and the learning progression from novice to expert.

The results of the descriptive statistics demonstrate a significant positive impact of trial exposure on decision-making progression, evidenced by a coefficient of 0.086 ($P < 0.05$). A mixed linear model regression confirms the effect of trials on decision-making and further reveals a variance of 0.007 in the random group effects. This indicates that while there are differences between groups, these differences are relatively small, suggesting that the trials themselves predominantly explain the variability in decision type.

Threshold analysis using ordered logistic regression reveals significant transition thresholds. The transition from novice to intermediate has a significant threshold of 0.88 ($P < 0.05$), indicating a challenging progression to higher decision-making skills. In contrast, the transition from intermediate to expert shows a significantly lower threshold of 0.1 (P = 0.021), suggesting it is easier to progress from intermediate to expert than from novice to intermediate.

4.1.4. The LLM-ACTR *Framework*

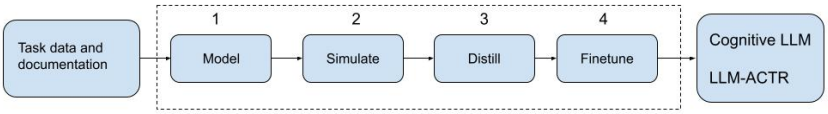

Figure 6. Decision augmentation using a neural-symbolic cognitive architecture approach. (1) Tasks are modeled with cognitive architecture. (2) Cognitive model used to run stochastic simulation of task at scale. (3) Synthetic data are distilled from simulation and combined with prompt requests. (4) A fine tuning pipeline is used to calibrate open source LLM to perform decision augmentation for task in exercise.

Figure 6 illustrates the approach to creating LLM-ACTR, which begins with the collection of task data and documentation. The task procedures are then modeled using ACT-R, employing stochastic simulations to analyze these tasks on a large scale. After the simulation phase, the generated synthetic data is semi-automatically distilled and combined with prompt requests. This data is subsequently used to infuse into an open source LLM through fine-tuning, resulting in a type of cognitive LLM, named LLM-ACTR.

Selecting Salient Decision Information ACT-R traces capture cognitive reasoning steps in real-time. These traces log the operations executed by various modules at each discrete decision point, including the activation of the goal module, the use of the imaginal buffer for accessing working memory, procedural memory matching and firing, utility updating driven by reinforcement learning, etc.

Preserving information from ACT-R model's decision-making traces poses challenges. A single decision-making round can generate a vast number of traces, each timestamped as frequently as every 5 milliseconds. Deciding which lines to select—or whether to preserve all lines—requires a balance between minimizing information loss and balancing computational costs. The rationale for choosing outputs from specific modules as reliable sources within the decision representation lies in their clear correspondence

to deterministic cognitive processes. The rationale for preserving all traces involves processes of semantic embedding extraction and dimensional reduction.

The information used to augment decision-making in this study focuses on distilling macro-level cognitive processes related to executive function[43], capturing the evolution of decision-making results across trials and how decisions adapt through learning and experience. Furthermore, the decision actions are categorized into strategy levels (novice, intermediate, expert), reflecting the learning phases.

Learning an Embedding Space of Decision Traces The next step involves converting the traces into tensors that the LLM can process. This study explores two approaches: one uses selected traces, and another uses full traces.

The selected traces are components distilled from macro-level cognitive processes related to executive function. This process requires human involvement to log decision results and strategy traces, which are then numerically encoded. For instance, '0' represents a decision for reduced time in pre-assembly section, and '1' for assembly. These data are subsequently fed into the neural network as single vectors.

In contrast, the holistic traces approach (see Figure 5a) retains both macro- and micro-level cognitive processes, with the latter including meta-cognition [44]. Metacognition involves an awareness and understanding of one's own cognitive processes, as exhibited through model traces that demonstrate the use of the imaginal buffer for accessing working memory, procedural memory matching and firing, headcount cost analysis, and the assessment of strategy effectiveness.

The investigation begins with the transformation of full traces from VSM-ACTR, representing both cognitive and metacognitive processes, into a format that balances information retention with computational efficiency. Cognitive reasoning traces for each task are processed through a sentence transformer to obtain semantic embeddings for each time stamp. A Sum of Ranked Explanatory Effects (SREE) analysis is then applied to determine the number (N) of principal components that account for at least 70% of the variance. Finally, these embeddings are reduced to N dimensions using Principal Components Analysis (PCA)[45].

Injecting Decision Information into LLM*s* With the VSM-ACTR model, which represents human-like cognitive reasoning in repeated decision-making tasks, this section outlines the experimental settings for fine-tuning of the `LLM-ACTR` framework. Fine-tuning, sometimes referred to as transfer learning, involves optimizing all model weights for the given task. The process includes parsing consistent template prompts that reflect the decision making task into an open-source LLM, aligning the task for the cognitive model using the LLM as the base model to access the last hidden layer and obtain masked embeddings, constructing a classification layer with softmax activation on top of the base model, using targets containing the salient decision representations of the cognitive model and features from the masked embeddings of the base LLM, and fine-tuning the LLM for classification using the LORA method. The key points are: (1) The targets decode the salient decision information from the cognitive model. (2) Use the final layer of contextualized embeddings in transformer-based LLMs, generated through the attention block mechanism. The attention block, a key feature of transformers, distinguishes them from other architectures like recurrent neural networks [46]. It creates embeddings that capture the in-context meaning of tokens by recombining them with other tokens' embeddings. Successive attention blocks further refine these embeddings, producing mul-

tiple layers of abstraction. The final layer, a blend of these refined embeddings, is used in this pipeline because it offers the richest semantic information while balancing minimal information loss and reduced computational costs for fine-tuning. (3) Use Low-Rank Adaptation (LoRa) for its efficiency in fine-tuning, reducing the computational resources and time required while maintaining high model performance [47].

4.1.5. Experiments

Problem Setting As an instantiation of the problem definition, above, our manufacturing line has two sections with potential defect sources: pre-assembly and assembly. Pre-assembly takes 40 seconds with an OEE rate of 88%, while assembly takes 44 seconds with an OEE rate of 80.1%. To reduce total assembly time by 4, we must identify which section can be shortened with minimal defect increase. We note that reducing cycle time will also lead to an increase in headcount costs.

Implementation Details The LLAMA-2 13B model was chosen as the foundation for this research because of its demonstrated effectiveness and efficiency in NLP tasks (Huang et al., 2024). As a state-of-the-art large language model, LLAMA has been trained on trillions of tokens from publicly available datasets. Unlike other transformer-based models such as the GPT family, which can only be accessed at the user's end, LLAMA 's architecture, including its pre-trained weights, is fully accessible. Furthermore, its proven capability to extract the last hidden layer for predicting behavioral discrepancies has been provided[48]. These attributes collectively establish LLAMA -2 13B as an optimal choice for this study.

To determine the dataset size that can effectively perform the task while balancing efficacy and resource limitations, we referred to [49], who showed evidence that LLAMA -2 13B achieves F1 scores above 0.9 in resource-limited text classification tasks, with datasets as 1,000 rows per class. Based on this, we developed the dataset size for fine-tuning as *N* (number of classes) * 1,000. The ACT-R dataset for binary decision-making classification contains 2,012 decision-making trials, Obtained by running the developed ACT-R model across 32 problem sets, each ACT-R persona was run for 15-16 trials until more stable expert behavior was achieved [42].

Baseline Models This study compared the goodness-of-fit and prediction accuracy of the resulting models using holdout data against two baselines: a random guess model and LLAMA without fine-tuning, obtained by reading out log-probabilities of the pre-trained LLAMA .

A random guess model serves as the most basic form of chance level baseline and represents the simplest hypothesis for model comparison. In psychological interdisciplinary experiments, control conditions often employ random responses to distinguish the effects of treatment from chance [50]. This approach allows assessing the extent to which decisions are influenced by knowledge versus being purely stochastic.

On the other hand, using LLAMA without fine-tuning as a baseline provides a reference point to measure the impact of fine-tuning on the model's performance. This comparison reveals how much the model 'learns' from the fine-tuning process compared to its generic, pre-trained state.

Research Questions Based on our framework's components, we identify a set of research questions that we answer through experiments.

1. *What are the properties of a useful neural network representation of the decision-making process in Cognitive Architectures?*

Answering this question sets the groundwork for developing a context-aware domain knowledge base for augmenting decision-making in LLMs.

2. *What level of complexity in behavior representation can* LLM*s effectively capture?*

Previous research has used LLM conceptual embeddings to predict human behavior based on past behavioral studies [51], confirming LLMs' ability to replicate known human patterns. However, high costs and extensive data collection efforts limit this method. By incorporating cognitive model simulations, the study seeks to address these limitations and broaden the investigation to determine the extent to which LLMs can reproduce decision-making knowledge. This will, in turn, help define the depth of decision-making domain knowledge that can be effectively integrated with the innate learning capabilities of LLMs.

3. *Can we inform the* LLM *with knowledge about the reasoning process of the cognitive architecture?*

Inspired by previous works on knowledge-injection [52,53], answering this question offers insights into knowledge transfer from domain-specific bases to LLMs and evaluates its impact on performance in holdout tasks. The method for addressing RQ1 was introduced in the first two sections of our approach framework.

Feature Extraction for Behavior Prediction To answer RQ2: What level of complexity in behavior representation can LLMs effectively capture? Building on previous research that used conceptual embeddings from LLMs to predict human behavior with historical behavioral data [51], we adopted the same method of LLM feature extraction for behavior prediction [54]. We created datasets consisting of last contextual embeddings as features and the corresponding different levels of VSM-ACTR decision actions representations as targets. We obtained embeddings by passing prompts that included all the information that VSM-ACTR had access to on a given trial through LLAMA and then extracting the hidden activations of the final layer, as shown in Figure 5b.

The first dataset used features extracted from prompts identical to the VSM-ACTR task, with targets being the VSM-ACTR decision-making results, where '0' indicates reduced time in preassembly and '1' indicates assembly. The second dataset's prompt template added an explanation of the strategy adopted by VSM-ACTR and used compound targets comprising both the decision-making results and the strategies reflecting the learning trajectory (novice, intermediate, and expert). The targets were encoded as follows: 0, 1, and 2 for preassembly choices using novice, intermediate, and expert strategies, respectively, and 3, 4, and 5 for assembly choices following the same pattern. With these two datasets, we fitted a regularized logistic regression model using 10-fold cross-validation for dataset 1 and multinomial regression using 10-fold cross-validation with L2 regularization for dataset 2. Model performance was assessed by measuring the goodness of fit through negative log-likelihood (NLL) and the predictive accuracy of hold-out data.

Fine Tuning for Knowledge Transfer To answer RQ3: whether LLMs can be informed with knowledge about the reasoning processes of cognitive architecture—we use the fine-tuning approach of `LLM-ACTR` Framework. The fine-tuning process employs Cross-

Entropy as the loss function and uses Adam optimization. Training involves a train test split of 0.2 and uses a batch size of 5 for both training and validation phases. The learning rate is set to 1e-5, with the training spanning across 10 epochs. To ensure regularization and prevent overfitting, a weight decay of 0.01, and a dropout of 0.5 are applied, and gradient accumulation is set to 2. Last but not least, gradient clipping is employed to maintain a maximum gradient norm of 1.0 for gradient explosion control. We evaluate the model fitting and generalization quality using training loss and validation loss across epochs, then compare the goodness of fit and prediction accuracy of the hold-out data against the baseline models.

4.1.6. Results

Finding Useful Decision Process Embeddings The approach of distilling macro-level cognitive processes related to executive function captures the evolution of decision-making results across trials and how decisions adapt through learning and experience, all represented as a sequential single vector. This format facilitates ease of use for downstream tasks involving knowledge transfer. However, this method retains only partial cognitive decision-making knowledge.

In contrast, the holistic semantic preservation approach encompasses both macro and micro-level cognition processes. However, the embeddings produced vary in shape due to the individual differences in traces originating from stochastic simulations. They cannot be directly fed into neural networks for downstream tasks. Nevertheless, the first two principal components of the reduced embeddings, which correspond to the semantic mapping of ACT-R's components—including procedural, imaginal, goal knowledge, utility updating, and decision-making—are detailed in Figure 6.

The MANOVA analysis was conducted to assess the overall effect of the independent variables, which include label categories or ACT-R components, on the combined dependent variables—components of reduced embeddings. This analysis reveals a significant relationship with the semantic mapping of ACT-R's components. For instance, the extremely low Wilks' lambda value (0.0004) suggests that the label or ACT-R component categories explain nearly all the variance in the dependent variables, indicative of a strong group effect. The statistical tests applied—Wilks' lambda, Pillai's trace, Hotelling-Lawley trace, and Roy's greatest root—all demonstrate strong significance, as evidenced by the extremely low p-values across all tests. These findings highlight that the principal components retained in the PCA successfully capture the essential variance related to these cognitive processes.

This result validates that our semantic abstraction method has the potential to retain the maximum semantics of neural symbolic representations at a minimal computational cost. However, further work is required to address the issue of ragged tensors for downstream tasks.

In a preliminary experiment, we addressed the issue of ragged tensors by employing padding with value imputation. We then integrated the 240 full cognitive reasoning traces from the VSM-ACTR model with LLM using embedding concatenation and conducted feature extraction for behavior prediction. Specifically, we transposed the reduced embeddings from each cognitive model run into a (1, X) dimension tensor and subsequently concatenated this with the LLM's last contextual embedding from the same prompt. These concatenated embeddings served as resources for predicting decision-making within the VSM-ACTR model. The prediction targets were multifaceted, includ-

ing both the decision-making results and the strategies used. The results showed no significant improvement in prediction accuracy with concatenated embeddings compared to using LLM embeddings alone.

One possible explanation is the relative scale of the VSM-ACTR reduced embeddings compared to those of LLAMA , which is disproportionately small (1:10). Consequently, the LLAMA embeddings may dominated the decision-making process within the model due to their larger scale. A potential solution could be to generate longer VSM-ACTR model traces, including tenfold more decision-making trials, thereby enhancing the scale and variability of its features.

Also, the method we use to handle ragged tensors—padding followed by value imputation—could potentially dilute the VSM-ACTR embeddings and reduce their accuracy. Finding an alternative method to preserve the full embeddings from VSM-ACTR may potentially improve the results.

Lastly, the limited dataset size could be influencing the results. The preliminary test used only 240 complete traces. Expanding the dataset may provide more insights into the performance of the proposed approach.

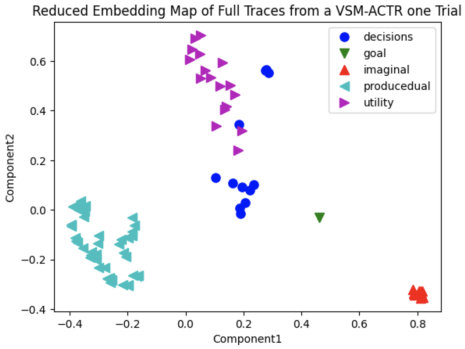

Figure 7. ACTR embedding mapping

Assessing Behavior Complexity Captured by LLMs Table 1 shows that LLM-ACTR captures a single facet of decision-making, achieving an average accuracy of 0.64 across 10 validation folds in the holdout task. When decision-making targets involve multiple facets—encompassing both choices and strategies that shape the learning trajectory—the accuracy decreases to 0.42. While this reduction suggests that capturing complex decision-making processes is less accurate, the results still show promise in handling these complexities. However, the Negative Log-Likelihood (NLL) reveals greater predictive uncertainty for multifaceted decision-making processes, as evidenced by a significantly higher NLL of 1.18 compared to 0.65 in single-facet scenarios.

Table 1. Evaluation for Single and Multi Facets Targets

Target Type	NLL	Accuracy
Single Facet Target	0.63	0.64
Multi Facets Target	1.18	0.42

Table 2. Comparison of VSM-ACTR with baselines

Model	NLL	Accuracy
Chance-level	0.6931	0.4826
LLAMA	1.1330	0.3564
LLM-ACTR(ours)	**0.6534**	**0.6576**

Injecting LLM*s with CA Decision Process* We first report training and validation losses, across 10 epochs, to reveal the fine-tuned model's learning and generalization behavior. Initially, the training loss begins at approximately 0.73, with a slight fluctuation observed in subsequent epochs, peaking around epoch 2 and showing a notable dip at epoch 7. In contrast, the validation loss starts at around 0.64 and remains remarkably stable throughout the epochs. This consistency in validation loss, coupled with a generally downward trend in training loss after its initial variations, suggests that the model is learning effectively. The overall trend indicates an improvement in model performance over time, reflecting its capability to generalize well on unseen data.

We then report the comparison of the `LLM-ACTR` with the baseline models on goodness of fit using negative log likelihood (NLL) and accuracy score for hold-out data. The `LLM-ACTR` model demonstrates significantly better performance across all metrics compared to the LLAMA -only model, highlighting its effectiveness in decision-making tasks involving sequential cognitive reasoning. Additionally, the LLAMA -only model performs worse than the chance-level model. This underscores the necessity of fine-tuning pre-trained language models like LLAMA to adapt them to specific human-aligned repeated decision-making tasks.

4.1.7. Discussion

The results shown in the previous section support our hypothesis that task-specific semantic interpretation enables cognitive reasoning mechanisms in AI. Our experiments demonstrate the benefits of using neuro-symbolic architectures to bridge the gap between neural pattern recognition and cognitive reasoning in AI. By leveraging task-specific semantic interpretations through cognitive model integration, we enable enhanced decision-making capabilities. The results validate the hypothesis that cognitive reasoning can be achieved using mechanisms similar to executing metacognitive rules captured within the parametric memory of foundation models, providing a plausible pathway for future developments in cognitively-inspired AI.

*4.2. **Mapping neural representations to cognitive representations, followed by instruction-based reasoning***

In the previous experiment, we adopted a design inspired by the CMC by compressing cognitive reasoning traces into neural representations. In this section, we explore an alternative approach, utilizing various neural network foundation models enhanced with external knowledge sources for perception and grounding. This is followed by a cognitive reasoning framework, also inspired by the CMC.

4.2.1. Preliminaries - Traditional AI components and the CMC

Figure 8 illustrates how traditional AI components correspond to the CMC components introduced in Section 2. Neural network-based processing methods are typically employed for perception, transforming raw data into abstractions that can be utilized by other CMC components (e.g., converting raw text in documents into noun phrases and other grammatical or syntactic structures).

The long-term memory stores "rules," which can be understood as sets of abstractions that, when evaluated, lead to other abstractions (e.g., if a text chunk is identified

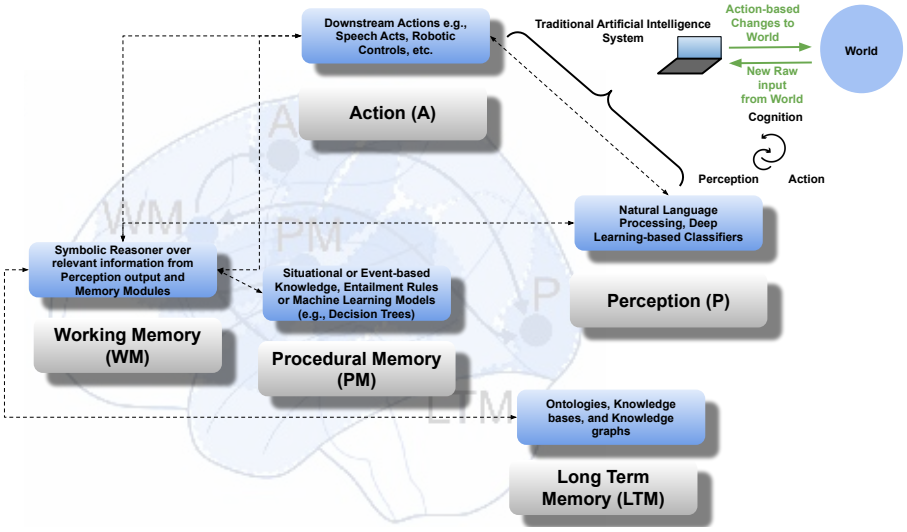

Figure 8. The common model of cognition [14]. The figure shows how traditional components of an AI system map to the various components of the CMC.

as a noun phrase, the next chunk is likely to be a verb phrase). Declarative knowledge refers to the semantics assigned to these abstractions in the procedural memory, often derived from curated knowledge bases, graphs, and ontologies (e.g., a noun phrase is composed of parts-of-speech such as nouns, adjectives, etc.). The working memory integrates the outputs of perception, procedural knowledge, and declarative knowledge to perform reasoning. For instance, if the current input is classified as a noun phrase, a rule in the procedural memory may suggest that the next chunk is likely to be a verb phrase.

4.2.2. Experimental Setup

The experimental setup focuses on activity recognition in egocentric scenes, which involves understanding specific actions within first-person perspectives. These scenes present a significant challenge due to the unpredictable behaviors often observed. For instance, during a cooking task, an individual may switch to unrelated activities, such as checking emails while waiting for food to boil. This introduces added complexity to the activity distribution within the scene.

Motivation As illustrated in Figure 8, the perception and cognition stages of the perception-cognition-action cycle involve an interplay between declarative, procedural, and working memory. This interaction builds a comprehensive representation of activity understanding, which is essential for downstream tasks. For example, in cooking-related activities, declarative memory may store factual knowledge, such as "foods contain salt," potentially supplemented by external knowledge sources. Procedural memory, typically a set of reasoning rules, could include how to answer specific video-related queries based on facts stored in declarative memory. Working memory, with limited capacity, draws on both declarative and procedural memory to execute the necessary reasoning steps during runtime. To evaluate the system's activity understanding capabilities, we designed two tasks aimed at monitoring how the system's activity representations are maintained across these memory structures.

4.2.3. Dataset-EGO4D

The dataset utilized in our experiments is EGO4D, a comprehensive collection of egocentric videos [55]. It consists of 3,600 hours of densely narrated video content, accompanied by detailed human annotations. The dataset covers a variety of scenarios—including household, outdoor, workplace, and leisure settings—recorded across 74 locations in 9 different countries. The video segments include supplementary data such as audio, 3D environmental meshes, eye gaze tracking, stereo video, and synchronized footage from multiple egocentric cameras capturing the same event. Additionally, this dataset includes challenges related to episodic memory recall and future event forecasting, with ground truth annotations provided. In the context of EGO4D, the queries often focus on atypical activity patterns. For example, a query related to "headphones" may need to be inferred by identifying appropriate video frames, such as those depicting a person multitasking by answering emails while cooking.

4.2.4. Methodology - Simulating the Perception-Cognition-Action Cycle in Egocentric Activity Recognition

To simulate the perception-cognition-action cycle in our experiment, we iterate between two tasks:

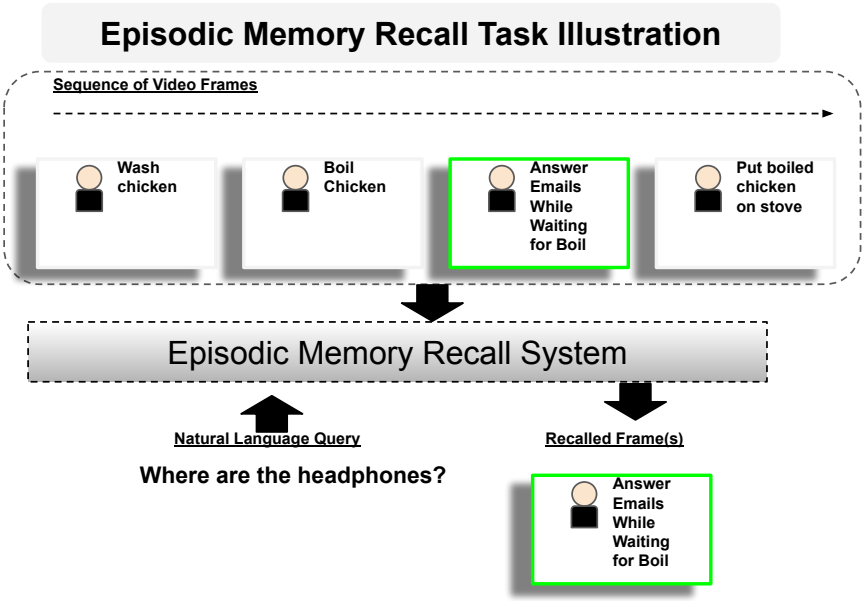

Figure 9. **(a)** Task 1 - Episodic Memory Recall.

Task 1 - Episodic Memory Recall In this task, a video scene and a natural language query are provided as inputs. The system is expected to output a sequence of frames from the video that likely contain the answer to the query, effectively "recalling" specific segments or "episodes" of the video. This process requires the system to establish and retrieve memory representations of the query and the video scene, particularly their

relationship in the context of the video's activity patterns. Figure 9 illustrates the inputs and outputs expected for a system performing this task. Notably, the query may involve complex associations, such as asking about headphones in relation to an unusual activity, like answering emails while boiling water. This presents a significant challenge for an AI system to interpret and infer.

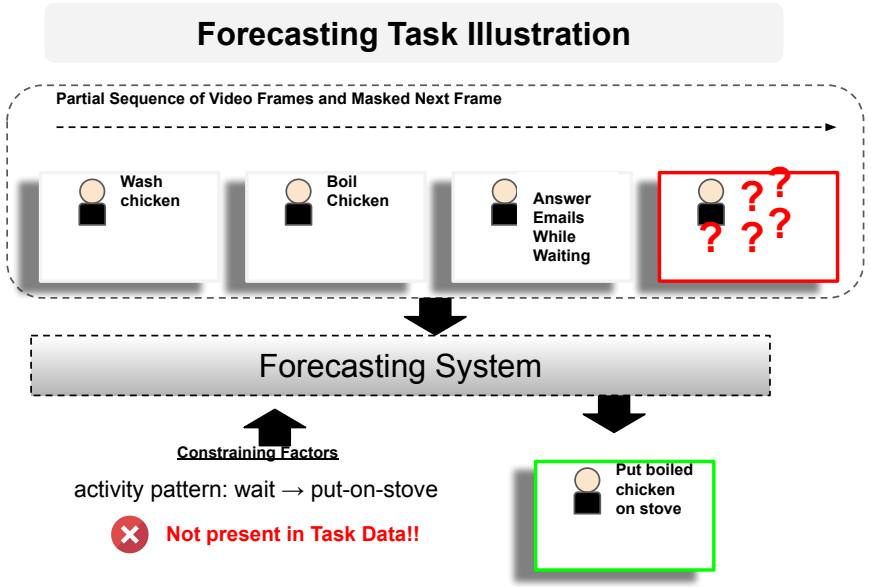

Figure 10. **(a)** Task 2 - Forecasting.

Task 2 - Forecasting In this task, a partial sequence of frames from a video is given as input, and the system must predict the action category for the next frame. Solving this task requires the system to make informed and constrained predictions about the possible upcoming actions, despite the potentially large number of possibilities. The quality of the memory structures, particularly in capturing activity patterns, is critical to the success of the prediction. Figure 10 illustrates this challenge, showing how the space of potential next actions can quickly become overwhelming. Typically, some external constraint—beyond the task dataset—is required to manage this complexity, and memory structures play a key role in applying these constraints.

The Perception-Cognition-Action Cycle for Activity Understanding Our proposed approach uses the CMC and components as scaffolding and maps foundation models as needed across the components. Figure 11 illustrates the approach with an example query and video (inputs do not contain the query in the forecasting task and only a partial clip of the video). We note that the system is *language-centric*, i.e., the task execution takes language-based inputs and produces language-based outputs. All other modalities of information are, therefore, first converted to natural language (text) before processing by the system.

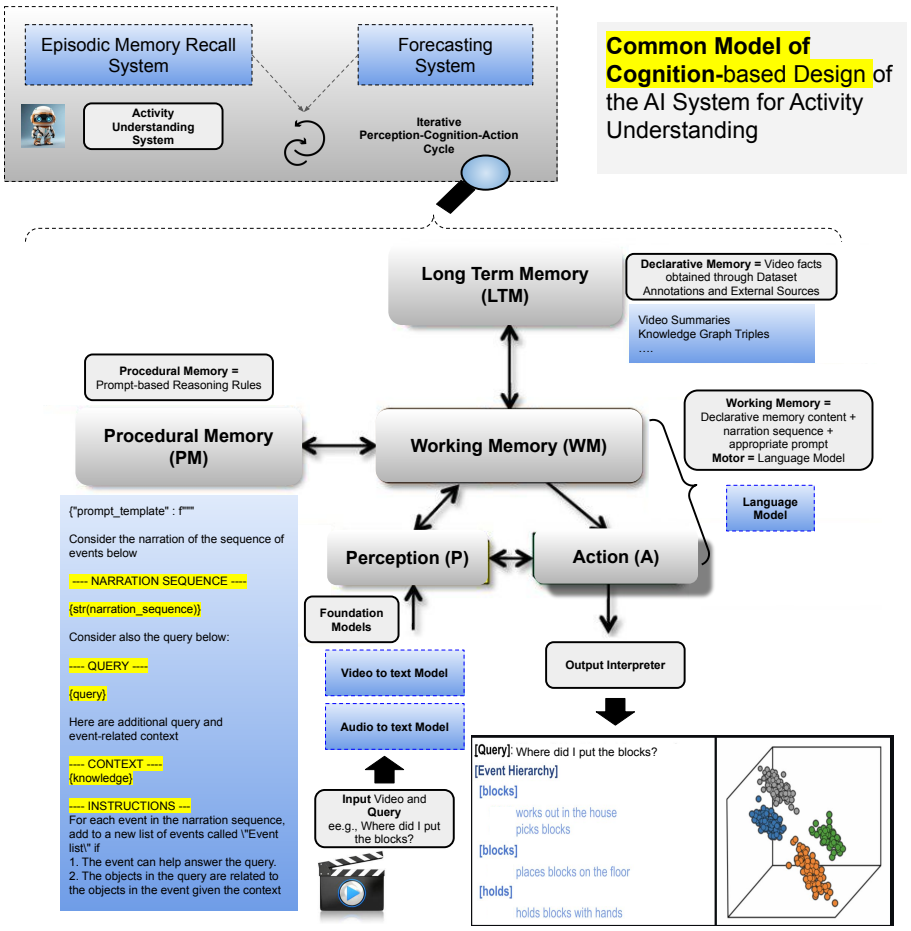

Figure 11. CMC-inspired solution approach and system design.

Why a language-centric approach? We choose a language-based core (representation and reasoning) for our approach for three reasons: (a) Language models are the most extensively studied foundation model, leading to better ease of access, software availability, and widespread methods for achieving relatively lower computation footprints, a key consideration for developing solutions to resource-bounded tasks. (b) Multimodal foundation models are still in their nascent stages. Therefore, multimodal model pipelines typically consist of projection to common representation spaces, and for reasons mentioned in (a), we choose language to be such a representation space. (c) Foundation models are prone to generating erroneous outputs, e.g., hallucinations, and augmenting the model with common sense knowledge is often beneficial in mitigating such errors. Most large-scale and publicly available knowledge bases are based on language, making language an attractive representation choice for augmenting language models with information from knowledge bases. The details of the individual CMC components are:

1. **Perception** The perception module takes the raw input, in this case, the video and the query, and transforms the information in the video, both the visual in-

formation and audio modalities, into text that consists of obtaining a "narration sequence" – a frame-by-frame narration of the video. The task Figures 9 and 10 show examples of such a narration sequence, i.e., the list [*Wash chicken, boil chicken, answer emails while waiting, and put boiled chicken on stove, ...*].

2. **Procedural Memory** In prototypical cognitive systems, procedural memory essentially consists of a set of rules or instructions that tell the system how to reason given an objective and the contents of the working memory. In this work, since the reasoner is language-based, the procedural memory contains prompts that guide or *instruct* the language model's reasoning.
3. **Declarative Memory** The declarative memory consists of language-based video annotations, such as video summaries and information sourced in knowledge graphs (verbalized using a verbalizer prompt). This information is meant to augment the narration sequence information and provide additional context to the reasoner.
4. **Working Memory** The working memory relies on outputs from the perception component, and the procedural and declarative memories. It is instantiated during reasoning to process the narration sequence, extracting task-related outputs by using relevant declarative memory content and procedural memory prompts. Importantly, the working memory is limited by the context window size of the reasoning LLM, a critical resource constraint. This CMC-inspired design emphasizes creating efficient representations and instructions in declarative and procedural memories to optimize reasoning within these constraints.
5. **Motor Module** The motor module consists of the language model that reasons about the task output given the contents of the working memory: in this regard, such module is responsible for speech acts. Accordingly, we augment the motor component with an output interpreter module that constructs a mapping of the combined contents of the working memory and predicted task outputs to an embedding space for visualizing task-related activity patterns.

4.2.5. Experimental Configuration

In this section, we describe the experimental configuration in detail. The CMC components are standard across the episodic memory recall and forecasting tasks, with only the input and output capture being slightly different (as shown in figures 9 and 10). The episodic memory recall task uses a video sequence as input and expects a video frame as a response to a natural language query, whereas in the forecasting task, the input is a partial sequence of contiguous frames, and the output is predicted action categories for the next frame. We now describe the details of the CMC components.

1. **Perception** The ground-truth annotations for the EGO4D dataset already consist of narration sequences obtained using model-based translations from visual and audio inputs. These annotations also contain the video clip time slice corresponding to each narration segment.
2. **Procedural Memory** The procedural memory consists of prompts (such as the one shown in Figure 11) for reasoning about events and actions in the video given narration sequences and declarative memory information.
3. **Declarative Memory** For declarative memory, the EGO4D dataset includes annotations for video summaries. Additionally, we utilize triples from the Common

Sense Knowledge Graph (CSKG) [56]. The triple extraction process generally follows a three-stage pipeline: (i) Identifying action phrases in the input query and narration sequences using the syntax parser from the Flair library [57]. (ii) Performing keyword-based triple extraction by matching individual words in the action phrases (using the word tokenizer from nltk) [58]. (iii) Verbalizing only the relevant subset of triples through a language model based on a given prompt. We also employ a third type of declarative memory, referred to as "internal" knowledge. This is derived by querying the language model to further analyze the input query and narration sequences, with the aim of retrieving relevant information embedded within its parametric knowledge. The various prompts used are available at this Github repository.

Note: During the execution of the forecasting task pipeline, we also maintain a memory bank of videos in declarative memory – This bank is updated when the episodic memory pipeline successfully identifies the correct video frame associated with the query, with high accuracy. Figure 10 shows how a constraint specification is essential to assist the system with predicting plausible next-action categories. This memory bank (i.e., the action distributions across the videos in this bank) acts as such a constraint specification.

Metacognitive Instruction-based Prompting *The steps involving procedural memory construction from declarative memory elements, mainly the various types of knowledge sources, denote the core of the metacognitive rule construction, where we aim to accurately specify a robust semantic interpretation of the task in the prompt (seen as a rule) to engender cognitive reasoning by the foundation model during prompt execution.*

4. **Working Memory** The working memory is simply a *text string* that is a combination of the query (the input), the narration sequence (from the perception component), the context (from the declarative memory), and the appropriate prompt (from the procedural memory). The size of the *text string* is limited by the context-window lengths of the language models used in the motor component. In the forecasting task, the working memory involves an information retrieval mechanism to retrieve a set of top-k similar videos from the memory bank in the declarative memory to constrain the set of possible next-action categories (k is set to 3 in our experiments, and the retrieval method used is RAPTOR [59]).
5. **Motor** The language models that we experiment with are LLaMa 3-8b-8192, mixtral-8x7b-32768, and gemma-7b-it, accessed using the GROQ API. The 'x'b part of the name denotes the number of parameters in billions, and the third part of the name after the '-' denotes the context window lengths in terms of no. of tokens (each token is roughly equal to a word).

4.2.6. Results and Discussion

Episodic Memory Task Tables 3, 4, and 5 present performance across different declarative memory options. Due to the complex action distributions in EGO4D, such as "answering emails while wearing headphones during cooking," the internal knowledge within the LLM is less effective for inference. In contrast, human-annotated video summaries perform best, with CSKG triples nearly matching their accuracy. Additionally, models with larger context windows tend to perform better overall.

Table 3. Evaluation of Episodic Memory Task with Internal Knowledge.

LLM	Accuracy	Precision	Recall	F1-score
gemma-7b-it	0.3962	0.4999	0.3962	0.4363
LLAMA 3.1-8b	0.5869	0.6749	0.5869	0.6203
mixtral-8x7b	**0.6291**	**0.7444**	**0.6291**	**0.6736**

Table 4. Evaluation of Episodic Memory Task with Video Summary Knowledge.

LLM	Accuracy	Precision	Recall	F1-score
gemma-7b-it	0.5999	0.6999	0.5999	0.6444
LLAMA 3.1-8b	0.7962	0.8999	0.7962	0.8363
mixtral-8x7b	**0.7999**	**0.9999**	**0.7999**	**0.8888**

Table 5. Evaluation of Episodic Memory Task with CSKG Knowledge.

LLM	Accuracy	Precision	Recall	F1-score
gemma-7b-it	0.5749	0.5493	0.5899	0.6076
LLAMA 3.1-8b	**0.7523**	0.7886	**0.8274**	0.7000
mixtral-8x7b	0.7269	**0.8666**	0.7799	**0.8133**

Forecasting Task - In the forecasting task, Tables 6-8 demonstrate that, without clarity on plausible next-action distributions as constraints, the performance of the declarative knowledge sources is quite poor. An exception to this is the summary knowledge, which performs better as it is annotated by humans. However, when information about similar videos is retrieved from the memory bank, the results are notably improved, as evidenced by Table 9.

Table 6. Evaluation of Forecasting Task with Internal Knowledge.

LLM	Accuracy	Precision	Recall	F1-score
gemma-7b-it	0.21	0.23	0.2899	0.30
LLAMA 3.1-8b	0.223	0.26	0.265	**0.3**
mixtral-8x7b	**0.286**	**0.3**	**0.33**	0.21

Table 7. Evaluation of Forecasting Task with Video Summary Knowledge.

LLM	Accuracy	Precision	Recall	F1-score
gemma-7b-it	0.66	0.7333	0.78	0.69
LLAMA 3.1-8b	0.63	0.78	0.779	0.78
mixtral-8x7b	**0.75**	**0.93**	**0.81**	**0.87**

Table 8. Evaluation of Forecasting Task with CSKG Knowledge.

LLM	Accuracy	Precision	Recall	F1-score
gemma-7b-it	0.13	0.14	0.20	0.33
LLAMA 3.1-8b	0.23	0.27	**0.30**	0.23
mixtral-8x7b	**0.28**	**0.29**	0.258	**0.29**

Table 9. Evaluation of Forecasting Task with Information Retrieval

LLM	Accuracy	Precision	Recall	F1-score
gemma-7b-it	0.52	0.6212	0.55	0.49
LLAMA 3.1-8b	0.65	0.71	0.633	0.69
mixtral-8x7b	**0.68**	**0.733**	**0.6412**	**0.7111**

5. Conclusion and Validity of the Hypothesis Statement

This chapter introduces a novel CMC-inspired approach to foundation model-based neurosymbolic cognitive AI systems with improved cognitive reasoning. The experiments conducted show that either strategy—compressing cognitive representations for integration with neural networks or mapping neural to cognitive representations followed by instruction-based reasoning—yield promising results. A CMC-based framework improves cognitive reasoning in foundation models, particularly by enabling metacognitive instruction-following behavior, supporting the main hypothesis.

References

[1] Ce Zhou, Qian Li, Chen Li, Jun Yu, Yixin Liu, Guangjing Wang, Kai Zhang, Cheng Ji, Qiben Yan, Lifang He, et al. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. *arXiv preprint arXiv:2302.09419*, 2023.

[2] Devon Myers, Rami Mohawesh, Venkata Ishwarya Chellaboina, Anantha Lakshmi Sathvik, Praveen Venkatesh, Yi-Hui Ho, Hanna Henshaw, Muna Alhawawreh, David Berdik, and Yaser Jararweh. Foundation and large language models: fundamentals, challenges, opportunities, and social impacts. *Cluster Computing*, 27(1):1–26, 2024.

[3] Nanyi Fei, Zhiwu Lu, Yizhao Gao, Guoxing Yang, Yuqi Huo, Jingyuan Wen, Haoyu Lu, Ruihua Song, Xin Gao, Tao Xiang, et al. Towards artificial general intelligence via a multimodal foundation model. *Nature Communications*, 13(1):3094, 2022.

[4] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

[5] Marianna Nezhurina, Lucia Cipolina-Kun, Mehdi Cherti, and Jenia Jitsev. Alice in wonderland: Simple tasks showing complete reasoning breakdown in state-of-the-art large language models. *arXiv preprint arXiv:2406.02061*, 2024.

[6] Subbarao Kambhampati. Can large language models reason and plan? *Annals of the New York Academy of Sciences*, 1534(1):15–18, 2024.

[7] Ruoqi Shen, Sébastien Bubeck, Ronen Eldan, Yin Tat Lee, Yuanzhi Li, and Yi Zhang. Positional description matters for transformers arithmetic. *arXiv preprint arXiv:2311.14737*, 2023.

[8] Trieu H Trinh, Yuhuai Wu, Quoc V Le, He He, and Thang Luong. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482, 2024.

[9] Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623, 2021.

[10] Joshua Tennenbaum. AAAI-23 Invited Speaker Series. `https://aaai-23.aaai.org/invited-speakers/`, 2023. [Online; accessed 9-Sep-2024].

[11] Goonmeet Bajaj, Valerie L Shalin, Srinivasan Parthasarathy, and Amit Sheth. Grounding from an ai and cognitive science lens. *IEEE Intelligent Systems*, 39(2):66–71, 2024.

[12] Vipula Rawte, Swagata Chakraborty, Agnibh Pathak, Anubhav Sarkar, SM Towhidul Islam Tonmoy, Aman Chadha, Amit Sheth, and Amitava Das. The troubling emergence of hallucination in large language models-an extensive definition, quantification, and prescriptive remediations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2541–2573, 2023.

[13] Ameet Deshpande, Vishvak Murahari, Tanmay Rajpurohit, Ashwin Kalyan, and Karthik Narasimhan. Toxicity in chatgpt: Analyzing persona-assigned language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1236–1270, 2023.

[14] John E Laird, Christian Lebiere, and Paul S Rosenbloom. A standard model of the mind: Toward a common computational framework across artificial intelligence, cognitive science, neuroscience, and robotics. *Ai Magazine*, 38(4):13–26, 2017.

[15] Andrea Stocco, Catherine Sibert, Zoe Steine-Hanson, Natalie Koh, John E Laird, Christian J Lebiere, and Paul Rosenbloom. Analysis of the human connectome data supports the notion of a "common model of cognition" for human and human-like intelligence across domains. *NeuroImage*, 235:118035, 2021.

[16] Yann LeCun. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review*, 62, 2022.

[17] Lilian Weng. Llm-powered autonomous agents. *lilianweng.github.io*, Jun 2023.

[18] Allen Newell. *Unified theories of cognition*. Harvard University Press, 1994.

[19] Terrence C Stewart. *A methodology for computational cognitive modelling*. PhD thesis, Carleton University, 2007.

[20] J Cliff Shaw, Allen Newell, Herbert A Simon, and TO Ellis. A command structure for complex information processing. In *Proceedings of the May 6-8, 1958, western joint computer conference: contrasts in computers*, pages 119–128, 1958.

[21] Theodore Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas Griffiths. Cognitive architectures for language agents. *Transactions on Machine Learning Research*.

[22] Mudit Verma, Siddhant Bhambri, and Subbarao Kambhampati. Theory of mind abilities of large language models in human-robot interaction: An illusion? In *Companion of the 2024 ACM/IEEE Interna-*

tional Conference on Human-Robot Interaction, pages 36–45, 2024.

[23] Matthew A Kelly and David Reitter. Holographic declarative memory: Using distributional semantics within act-r. In *2017 AAAI Fall Symposium Series*, 2017.

[24] Amit Sheth and Kaushik Roy. Neurosymbolic value-inspired artificial intelligence (why, what, and how). *IEEE Intelligent Systems*, 39(1):5–11, 2024.

[25] Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. Meta-learning via language model in-context tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 719–730, 2022.

[26] Pascal Hitzler and Md Kamruzzaman Sarker. Neuro-symbolic artificial intelligence: The state of the art. 2022.

[27] Alessandro Oltramari. Enabling high-level machine reasoning with cognitive neuro-symbolic systems. In *Proceedings of the AAAI Symposium Series*, volume 2, pages 360–368, 2023.

[28] Artur d'Avila Garcez and Luis C Lamb. Neurosymbolic ai: The 3 rd wave. *Artificial Intelligence Review*, 56(11):12387–12406, 2023.

[29] Amit Sheth, Kaushik Roy, and Manas Gaur. Neurosymbolic artificial intelligence (why, what, and how). *IEEE Intelligent Systems*, 38(3):56–62, 2023.

[30] Iuliia Kotseruba and John K Tsotsos. 40 years of cognitive architectures: core cognitive abilities and practical applications. *Artificial Intelligence Review*, 53(1):17–94, 2020.

[31] John R Anderson. Act: A simple theory of complex cognition. *American psychologist*, 51(4):355, 1996.

[32] Sterling Somers, Alessandro Oltramari, and Christian Lebiere. Cognitive twin: A cognitive approach to personalized assistants. In *AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering (1)*, 2020.

[33] Mehdi Cina and Ahmad B Rad. Categorized review of drive simulators and driver behavior analysis focusing on act-r architecture in autonomous vehicles. *Sustainable Energy Technologies and Assessments*, 56:103044, 2023.

[34] Hao Chen, Shuang Liu, Liping Pang, Xiaoru Wanyan, and Yufeng Fang. Developing an improved act-r model for pilot situation awareness measurement. *IEEE Access*, 9:122113–122124, 2021.

[35] Noam Ben-Asher, Alessandro Oltramari, Robert F Erbacher, and Cleotilde Gonzalez. Ontology-based adaptive systems of cyber defense. In *STIDS*, pages 34–41, 2015.

[36] S. Wu, R. Ferreira, F. E. Ritter, and L. Walter. Comparing llms for prompt-enhanced act-r and soar model development: A case study in cognitive simulation. In *Proceedings of the 38th Annual Association for the Advancement of Artificial Intelligence (AAAI) Conference on Artificial Intelligence, Fall Symposium Series on Integrating Cognitive Architecture and Generative Models*, Arlington, Virginia, USA, 2023.

[37] B. J. Best and C. Lebiere. Teamwork, communication, and planning in act-r agents engaging in urban combat in virtual environments. In *Proceedings of the 2003 IJCAI Workshop on Cognitive Modeling of Agents and Multi-Agent Interactions*, pages 64–72, 2003.

[38] C. Gonzalez, J. F. Lerch, and C. Lebiere. Instance-based learning in dynamic decision making. *Cognitive Science*, 27:591–635, 2003.

[39] Sabine Prezenski, André Brechmann, Susann Wolff, and Nele Russwinkel. A cognitive modeling approach to strategy formation in dynamic decision making. *Frontiers in Psychology*, 8:1335, 2017.

[40] S. B. Blessing and J. R. Anderson. How people learn to skip steps. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 22(3):576, 1996.

[41] M. K. Martin, C. Gonzalez, and C. Lebiere. Learning to make decisions in dynamic environments: Act-r plays the beer game. In *Proceedings of the Sixth International Conference on Cognitive Modeling*, Pittsburgh, PA, 2004. Carnegie Mellon University/University of Pittsburgh.

[42] F. E. Ritter, M. J. Schoelles, K. S. Quigley, and L. C. Klein. *Determining the number of model runs: Treating cognitive models as theories by not sampling their behavior*, pages 97–116. Springer-Verlag, London, 2011.

[43] S. J. Gilbert and P. W. Burgess. Executive function. *Current Biology*, 18(3):R110–R114, 2008.

[44] T. O. Nelson and L. Narens. Why investigate metacognition? *Journal Name Here*, Volume Here(Issue Number Here):Page Numbers Here, 1994.

[45] H. Abdi and L. J. Williams. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):433–459, 2010.

[46] Alex Graves. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*, 2012.

[47] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and

Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[48] M. Binz and E. Schulz. Using cognitive psychology to understand gpt-3. *Proceedings of the National Academy of Sciences*, 120(6):e2218523120, 2023.

[49] Amit Kumar, Rakesh Sharma, and Punam Bedi. Towards optimal nlp solutions: Analyzing gpt and llama-2 models across model scale, dataset size, and task diversity. *Engineering, Technology & Applied Science Research*, 14(3):14219–14224, 2024.

[50] Jens Gaab, Joe Kossowsky, Ulrike Ehlert, and Cosima Locher. Effects and components of placebos with a psychological treatment rationale—three randomized-controlled studies. *Scientific Reports*, 9(1):1421, 2019.

[51] M. Binz and E. Schulz. Turning large language models into cognitive models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024. In press.

[52] Alessandro Oltramari, Jonathan Francis, Filip Ilievski, Kaixin Ma, and Roshanak Mirzaee. Generalizable neuro-symbolic systems for commonsense question answering. In *Neuro-Symbolic Artificial Intelligence: The State of the Art*, pages 294–310. IOS Press, 2021.

[53] Kaixin Ma, Jonathan Francis, Quanyang Lu, Eric Nyberg, and Alessandro Oltramari. Towards generalizable neuro-symbolic systems for commonsense question answering. *arXiv preprint arXiv:1910.14087*, 2019.

[54] Z. Hussain, M. Binz, R. Mata, and D. U. Wulff. A tutorial on open-source large language models for behavioral science. *PsyArXiv preprint*, 2023.

[55] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18973–18990. IEEE Computer Society, 2022.

[56] Filip Ilievski, Pedro Szekely, and Bin Zhang. Cskg: The commonsense knowledge graph. In *The Semantic Web: 18th International Conference, ESWC 2021, Virtual Event, June 6–10, 2021, Proceedings 18*, pages 680–696. Springer, 2021.

[57] Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. Flair: An easy-to-use framework for state-of-the-art nlp. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics (demonstrations)*, pages 54–59, 2019.

[58] Edward Loper and Steven Bird. Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, pages 63–70, 2002.

[59] Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D Manning. Raptor: Recursive abstractive processing for tree-organized retrieval. In *The Twelfth International Conference on Learning Representations*.