

Cognitive LLMs: Toward Human-Like Artificial Intelligence by Integrating Cognitive Architectures and Large Language Models for Manufacturing Decision-making

Siyu Wu^{a,*}, Alessandro Oltramari^{b,c}, Jonathan Francis^{b,c}, C. Lee Giles^a and Frank E. Ritter^a

^a College of Information Sciences and Technology, The Pennsylvania State University, University Park, PA USA

E-mail: {sfw5621, clg20, frank.ritter}@psu.edu

^b Bosch Center for Artificial Intelligence, Pittsburgh, PA USA

^c Carnegie Mellon University, Pittsburgh, PA USA

E-mail: {alessandro.oltramari, jon.francis}@us.bosch.com

Abstract. Resolving the dichotomy between the human-like yet constrained reasoning processes of Cognitive Architectures (CAs) and the broad but often noisy inference behavior of Large Language Models (LLMs) remains a challenging yet exciting pursuit, aimed at enabling reliable machine reasoning capabilities in LLMs. Previous approaches that employ off-the-shelf LLMs in manufacturing decision-making face challenges in complex reasoning tasks, often exhibiting human-level yet unhuman-like behaviors due to insufficient grounding. This present paper start to address this gap by asking whether LLMs can replicate cognition from CAs to make human-like decisions. We introduce *Cognitive LLMs*, which are hybrid decision-making architectures comprised of a CA and an LLM through a knowledge transfer mechanism *LLM-ACTR*. *Cognitive LLMs* extract and embed knowledge of CA's internal decision-making process as latent neural representations, inject this information into trainable LLM adapter layers, and fine-tune the LLMs for downstream prediction tasks. We find that, after knowledge transfer through *LLM-ACTR*, the *Cognitive LLMs* offers better representations of human decision-making behaviors on a novel Design for Manufacturing problem, compared to an LLM-only model that employs chain-of-thought. Taken together, the results open up new research directions for equipping LLMs with the necessary knowledge to computationally model and replicate the internal mechanisms of human cognitive decision-making. We release the code and data samples at <https://github.com/SiyuWu528/LLM-ACTR>.

Keywords: Cognitive architectures, Large language models

Introduction

Large Language Models (LLMs) have gained considerable popularity for a wide range of prediction and

decision-making tasks, spanning applications, such as robotics and control, neural question-answering, scene understanding, code generation, mathematical reasoning. LLMs are trained on massive datasets, can be used both as discriminative scoring functions as well as generative models, and their capacity allows them

*Work done during an internship at Bosch R&T center.

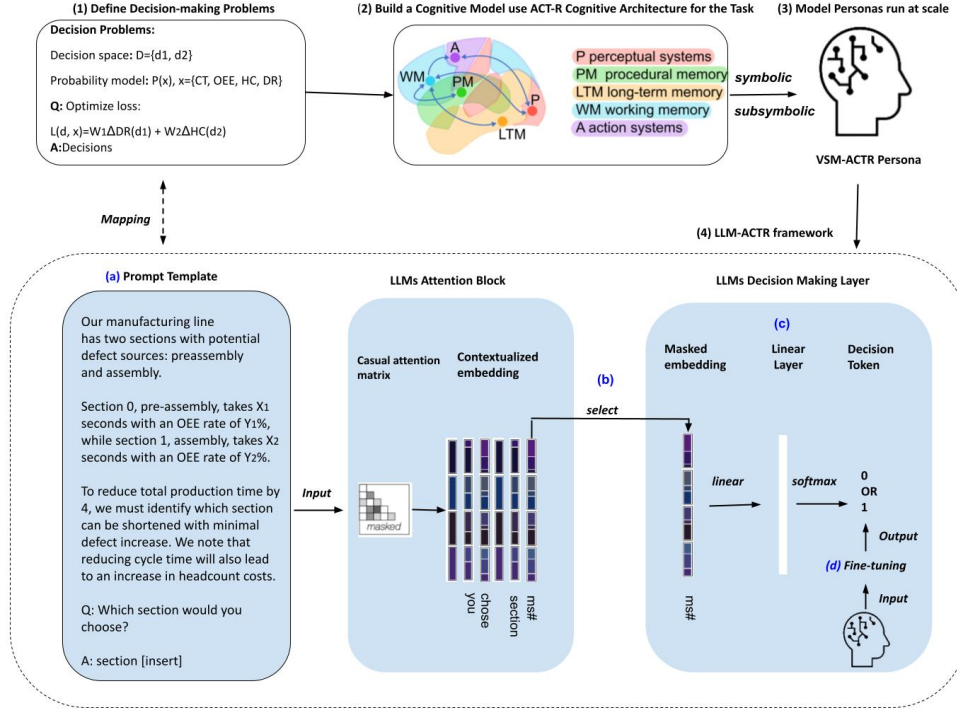


Fig. 1. Cognitive LLMs architecture, where CAs instruct LLMs for cognitive decision-making using *LLM-ACTR* knowledge transfer framework.

to accumulate and retain vast amounts of knowledge [7, 15, 20, 24, 34, 78]. Typical LLMs' use resembles *system-1 reasoning process* [30, 70], offering quick, intuitive responses for everyday tasks. And advancements in multi-agent LLM frameworks and emergent capabilities such as in-context learning [19, 20, 82] have pushed LLMs toward *system-2 reasoning process* [80], e.g., 'chain-of-thought' reasoning (CoT) [10], enabling more deliberate cognition for complex decisions [15, 85]. However, issues such as discrepancies in human-like reasoning [51], problems with insufficient grounding [91], and hallucination [16] persist. Specifically, when using off-the-shelf LLMs to augment decision-making in manufacturing, where the Value Stream Map (VSM) [65] with intertwined variables is vital for smart scheduling [69], plant managers often struggle with using LLMs' unhuman-like and noisy predictions [52] (also see Appendix: LLM Conversation Examples).

Toward trustworthy decision-making by LLMs in manufacturing, we ask whether LLMs can replicate cognition from Cognitive Architectures (CAs) to make human-like decisions. We propose *Cognitive LLMs* as an solution, which are hybrid decision-making ar-

chitectures comprised of a CA and an LLM through a developing knowledge transfer framework *LLM-ACTR*. CAs are codable computational frameworks designed to capture the invariant mechanisms of human cognition. These mechanisms include functions related to attention, control, learning, memory, adaptivity, perception, and action [47, 77]. Through CAs we can construct cognitive decision-making models that can store, retrieve, and process knowledge, e.g., [38, 54]. *Cognitive LLMs* extract and embed knowledge of cognitive model's internal decision-making process as latent neural representations, inject this information into trainable LLM adapter layers, and fine-tune the LLMs for downstream prediction tasks.

Cognitive LLMs (Fig. 1) begins with (1) defining decision-making problems from manufacturing management documentation, considering domain knowledge such as the VSM and human factors like feedback from plant managers; (2) We then use the representative cognitive architecture ACT-R [47, 66], widely used for understanding human cognition [5] and modeling human behaviors [6], to build a cognitive model. The model simulates human-like decision-making to address the defined problem. Techniques

such as ontology-based formalization and psychometrics are employed to model the symbolic components of the task, i.e., declarative and procedural knowledge, and to set the subsymbolic parameters, e.g., learning rate, similarity matching. (3) The cognitive model is then run at scale stochastically to collect cognitive decision-making reasoning stamps. Collected data are processed into vector embeddings using techniques such as tokenization and dimensionality reduction. (4) Lastly, Cognitive LLMs learn the embedded vectors of cognitive decision-making through the developing knowledge transfer framework *LLM-ACTR*. It leverages the strengths of both LLMs and CAs by using the natural language processing and generative capabilities of LLMs, complemented by the human-like learning and reasoning offered by CAs.

We present a case study of Cognitive LLMs in manufacturing decision-making. The task is associated with a key aspect of Design For Manufacturing (DFM): enhancing product development and optimizing production system performance by improving time efficiency and reducing headcount costs [81].

The present paper poses three research questions:

RQ1. *What are the properties of a neural network representation of the decision-making process in CAs?* Answering this question sets the ground for developing a context-aware domain knowledge base for augmenting decision-making in LLMs.

RQ2. *What level of complexity in behavior representation can LLMs capture?* Previous research used LLMs' conceptual embeddings to predict human-reinforced decisions [12], indicating that embeddings from LLMs could be trained to predict human-like behaviors. By incorporating more training sets using CAs, the study addresses the limitation of high data collection costs with human subjects and aims to broaden the investigation into the extent to which innate LLMs can learn human cognition.

RQ3. *Can we inform the LLMs with knowledge about the reasoning process of the CAs?* Answering this question offers insights into knowledge transfer from domain-specific bases to LLMs, and opens up new research directions for equipping LLMs with the necessary knowledge to computationally model and replicate the internal mechanisms of human cognitive decision-making.

The following sections are sequentially arranged as follows: related work; an explanation of Cognitive

LLMs, which comprises two components: the CA and its constructed cognitive model; the *LLM-ACTR* framework, which facilitates knowledge transfer using a developed domain knowledge base; and the experiments conducted to address the research questions, followed by the results, discussion, and implications.

Related Work

This section starts by integrating cognitive psychology principles into LLMs, along with decision intelligence in manufacturing and cognitive decision-making. It then highlights the domain limitations of these approaches. It concludes by discussing the current integrating of CAs and LLMs, and points out how our approach differs from others.

Relating Cognitive Psychology to Human-Like Artificial Intelligence

Human-like artificial intelligence (HLAI) has been a goal since the emergence of machines [56]. In recent years, the development of transformer-based LLMs has revolutionized HLA, demonstrating impressive human-level capabilities. However, LLMs sometimes fail to display human-like behavioral traits. Analyzing the areas where LLMs currently fall short in replicating human cognition and behavior highlights the problems in exhibiting human-level capabilities that are unhuman-like [21], including behavior discrepancies between LLM inference and human reasoning [11, 51], insufficient grounding [91], and hallucination [16].

The challenges mentioned have catalyzed an integration of cognitive psychology with LLMs, toward human-like trustworthy LLMs. Recent studies have used cognitive psychology experiments to investigate and comprehend behaviors in these models, focusing more on behavioral insights than on conventional performance metrics [11, 18]. In addition, the use of LLMs' neural representations has been applied in behavioral psychological science research, which involves and not limited to prompt engineering, feature extraction, and fine-tuning:

Feature Extraction. The process begins with passing text that mirrors a psychological experiment through the open-source LLM to capture contextualized embeddings from the final layer [36]. These embeddings can be employed in various psychological experiments

1 applications, such as predicting similarities between
2 personality constructs [2], choices in reinforcement
3 learning [12], or perceptions related to risk or health
4 [89]. For tasks that require sequence prediction, de-
5 coder models are preferred due to their larger size and
6 more extensive training data [36].

7 **Zero-shot and Few-shot Learning.** Zero-shot learn-
8 ing enables the creation of categorical or numerical
9 predictions, such as evaluating sentiments on social
10 media [22], without requiring training specific to the
11 task. Few-shot learning extends this concept by adding
12 minimal supervision, such as a small number of exam-
13 ple pairs, to improve the accuracy of the model.
14

15 **Fine-tuning.** Fine-tuning smaller LLMs for human-
16 like behaviors can achieve performance that matches
17 or exceeds that of larger models under zero- or few-
18 shot learning conditions [36]. This involves adjust-
19 ing model weights to improve task-specific outcomes.
20 For example, one study fine tuned BERT in zero-shot
21 learning to predict reinforcement learning behaviors of
22 human subjects [36]. However, the generalization of
23 this approach is challenged by the high cost of col-
24 lecting large cognitive psychological datasets involv-
25 ing human subjects.
26

27 *Common Model of Cognition, Cognitive* 28 *Architectures, and Cognitive Models* 29

30 Toward integrating human-like behavioral traits into
31 LLMs, we use a suite of tools rooted in the Com-
32 mon Model of Cognition (CMC) to provide a wider
33 range of tasks into the training dataset. CMC em-
34 bodies unified Theory of Mind [47, 58], a theoret-
35 ical framework that presents a model of human cog-
36 nition codified as a computational architecture. The
37 CMC is a brain-inspired framework validated by large-
38 scale neuroscience data. The CMC identifies core com-
39 ponents and processes fundamental to human cog-
40 nition, including memory, perception, motor actions,
41 and decision-making. The model assumes a cyclical
42 process where these components interact to produce
43 human behavior. The CMC includes a feature-based
44 declarative long-term memory, a buffer-based working
45 memory, a system for pattern-directed action invoca-
46 tion stored in procedural memory, and specialized sys-
47 tems for perception and action [71].
48

49 The CMC integrates essential features from various
50 CAs [5, 43, 44, 46], which propose a set of fixed mech-
51 anisms to model human behavior, functioning akin to

1 agents and aiming for a unified representation of the
2 mind. By using task-specific knowledge, these archi-
3 tectures not only simulate but also explain behavior
4 through direct examination and real-time reasoning
5 tracing.
6

7 Two representative cognitive architectures related to
8 the CMC are ACT-R and Soar [48]. Other CA could
9 also be chosen from a recent extensive review [43, 44],
10 as long as a trace is available. **ACT-R** is a theory of
11 simulating and understanding human cognition [6, 66],
12 through which we can construct models that can store,
13 retrieve, and process knowledge, as well as explain and
14 predict performance [14]. The two most commonly
15 used representations in ACT-R are declarative knowl-
16 edge and procedural knowledge. Declarative knowl-
17 edge consists of chunks of memory (e.g., the pro-
18 duction line comprises five sections), while procedu-
19 ral knowledge performs basic operations, moves data
20 among buffers, and identifies the next instructions to
21 be executed (e.g., lower defect rate will lead to higher
22 efficiency rate). **Soar**, on the other hand, is a gen-
23 eral cognitive architecture that provides a computa-
24 tional infrastructure that resembles the cognitive ca-
25 pabilities exhibited by a human [46]. It implements
26 knowledge-intensive reasoning that enables execution
27 of rules based on the context, and the capability to in-
28 tegrate learning into the intelligent agent using chunk-
29 ing or reinforcement learning. Soar’s general comput-
30 ing concept is based on objectives, problem spaces,
31 states and operators. Soar encompasses multiple mem-
32 ory constructs (e.g., semantic, episodic, etc.) and learn-
33 ing mechanisms (e.g., reinforcement, chunking etc.)
34

35 One primary difference between these two architec-
36 tures is that ACT-R was designed to model human be-
37 havior and has a track record of predicting human per-
38 formance and timing to the millisecond level. In con-
39 trast, Soar places less emphasis on replicating human
40 behavior and more on developing general agents with
41 cognitive capabilities [48].
42

43 *Decision Intelligence in Manufacturing* 44

45 Industry 4.0 aims to create ‘intelligent factories,’
46 where advanced manufacturing technologies facilitate
47 smart decision-making through real-time communica-
48 tion and cooperation among humans, machines, and
49 sensors [32]. One example of this is smart scheduling,
50 which employs advanced models and algorithms using
51 sensor data [69].

Decision intelligence [49] is a crucial component of smart scheduling and comprises three stages. **Decision support.** Machines provide basic tools to aid human decision-making, such as alerts, analytics, and data exploration. Here, the decisions are made entirely by humans. **Decision augmentation.** Machines take on a more proactive role in the decision-making process. They analyze data and generate recommendations and predictions for decision-makers to review and validate. Humans can base their decisions on these suggestions, or they can collaborate with the machine to refine the recommendations. **Decision Automation.** Machines handle both the decision-making and execution steps autonomously. Humans maintain a high-level overview, monitoring risks and unusual activities, and regularly review outcomes to enhance the system.

A value stream map (VSM) is a critical tool in manufacturing decision intelligence, functioning as a flowchart that visualizes and controls the production line [53]. VSM meticulously tracks metrics such as inputs, outputs, processes, overall equipment effectiveness (OEE), and cycle times (CT). However, plant managers encounter significant challenges when transitioning VSM in production management from decision support to decision augmentation. These challenges stem from the difficulty of applying VSM concepts to complex, real-world scenarios characterized by numerous intertwined variables [52].

Cognitive Decision Making

Representative CAs, e.g., Soar, ACT-R, have been used to build models that automate decision-making tasks, e.g., [38, 54]. Among them, the ACT-R cognitive architecture is applied to build models across psychology and computer science that are closely aligned with human behaviors. It has a track record of accurately predicting human performance and timing across a variety of tasks (see [63]), which meets our needs for developing synthetic agents that can provide human-like cognitive reasoning in learning and training environments.

The ACT-R modeling approaches include: (a) strategy or rule-based, where different problem-solving strategies are implemented through various production rules and successful strategies are rewarded [9, 87]; (b) exemplar or instance-based, which relies on past experiences stored in declarative memory to solve problems [28]; and (c) hybrid approaches that combine strategies and exemplars [64].

A few features distinguish the use of ACT-R in creating decision-making models that involve learning: **Modular design that mirrors the symbolic aspects of 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 can retrieve relevant memories and activate appropriate rules, ensuring both efficient and adaptive performance in decision-making tasks. It does so at a pace that mirrors human performance at the millisecond level.

However, ACT-R does not have LLM-like dialogic interaction with other ACT-R models, which limits their usability for decision-making. Intuitively, a solution could take the best of both CAs and LLMs, where ACT-R models serve as synthetic agents to instruct LLMs. They do this by providing knowledge of cognitive decision-making through LLMs' training, which includes aspects such as learning. The trained LLMs can then be generalized to unseen problems.

Integration of Cognitive Architectures and LLMs

Efforts have been made toward leveraging the strengths of both CAs and LLMs to create a more robust unified theory of computational cognitive models. Some approaches include using the implicit world knowledge of LLMs to replace traditional declarative knowledge mechanisms [86], employing Chain-of-Thought reasoning to enhance the symbolic mechanisms for procedural knowledge [41], and leveraging language models as external knowledge sources for cognitive systems, while exploring ways to improve the effectiveness of knowledge extraction [39].

Moreover, Sumers et al. [72] examines how principles from cognitive architectures can guide the design of LLM-based agent frameworks, demonstrating a comprehensive integration effort that spans from knowledge representation to interaction with the environment. Additionally, Sun [74] proposes a direction for creating computational cognitive architectures using dual-process models and hybrid neuro-symbolic methods. Using the Clarion CA [73] as an example, Sun illustrates the theoretical opportunities for incorporating

LLMs into Clarion’s modules of perception, memory, motor control, and communication, leveraging LLMs’ natural language processing and generalization abilities. This present study builds upon previous research; however, we have adopted a different perspective by leveraging CAs to ground the decisions of LLMs in a data-driven manner. We aim to examine the properties of a neural network representation of the decision-making process in CAs and investigate whether knowledge from CAs can be preserved in an embedding space and infused into LLMs through the transfer of learning.

Problem Definition: Design for Manufacturing

This paper presents a case study of training a cognitively inspired LLM for decision-making in the design for manufacturing (DFM) domain. We define the terminology that constitutes our decision-making problem. The DFM problem setting is a prototypical manufacturing production-line workflow, from supplier to customer, for which there exists a VSM (Fig. 2), which allows for tracking the efficiency at different sectors of the process and abstracts the overall problem for mathematical modeling and optimization. Decision candidates come from sectors such as Body Production, Pre-Assembly, Assembly. 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/or Mean Absolute Error (MAE).

Focused on maintaining stable output for manufacturing plants, we consider plant managers’ feedback alongside the VSM structure to define the decision-making problem that aim to reduce total production time while minimizing total defect rate increase (see Fig.1(1) Define Decision-Making Problems). When facing unseen DFM problems, which are yet constrained to fixed decision candidates and unknown decision metrics. Cognitive LLMs takes a natural language question prompt (see Fig.1(a) for Prompt Template), and outputs a binary decision (0 or 1) on which of two sectors, pre-assembly or assembly, requires a time reduction.

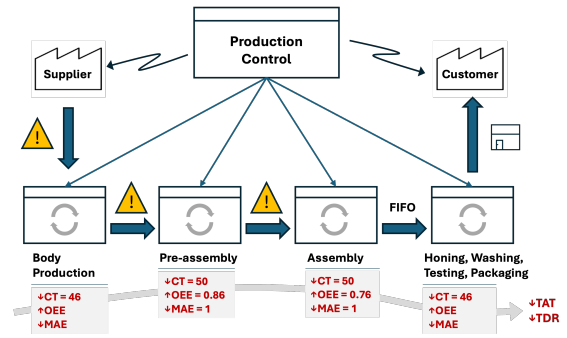


Fig. 2. A Value Stream Map of our manufacturing task process.

Cognitive LLMs: Hybrid Architectures for Human-Aligned Decision Making

Cognitive LLMs are comprised of a CA and an LLM through a developing knowledge transfer mechanism *LLM-ACTR*. Thus, we start by introducing the selected cognitive architecture ACT-R, then details about *LLM-ACTR*.

VSM-ACTR, A Human-Like Decision Making Cognitive Model

The ACT-R cognitive architecture was chosen to develop the cognitive model for our task because it has a track record of accurately predicting human performance and timing across a variety of tasks, which meets our need to develop synthetic agents with individual differences in learning and training, e.g., [54, 63]. We created the *VSM-ACTR* cognitive model, which is a rule-based ACT-R cognitive decision-making model for the DFM problem that implements multiple problem-solving strategies through a combination of production rules.

VSM-ACTR has incorporated the meta-cognitive processes that reflect on and evaluate the progress of chosen strategies—with an emphasis on headcount (manufacturing) cost evaluation, through a reward structure that enables a process akin to reinforcement learning. This system enables the model to dynamically assess the impact of decisions on headcount costs, computing a reward or penalty for each decision cycle. These rewards or penalties then dynamically adjust the utility of the productions associated with each decision-making cycle. This helps the model to exhibit a human-like learning progression.

Declarative Memory

VSM-ACTR integrates the prototypical decision process with insights into how cognitive models represent different levels of expertise, e.g., [55, 61], 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.

Production Rule Sets

Three sets of production rules represent the decision-making behaviors of novice, intermediate, and expert decision-makers. These sets comprise a total of 18 rules, each driven by goal-focused objectives across 20 states, covering actions such as choosing strategies, actions, working memory management, decisions, and evaluations.

We use the expert production rule set as an example (Fig. 3), 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

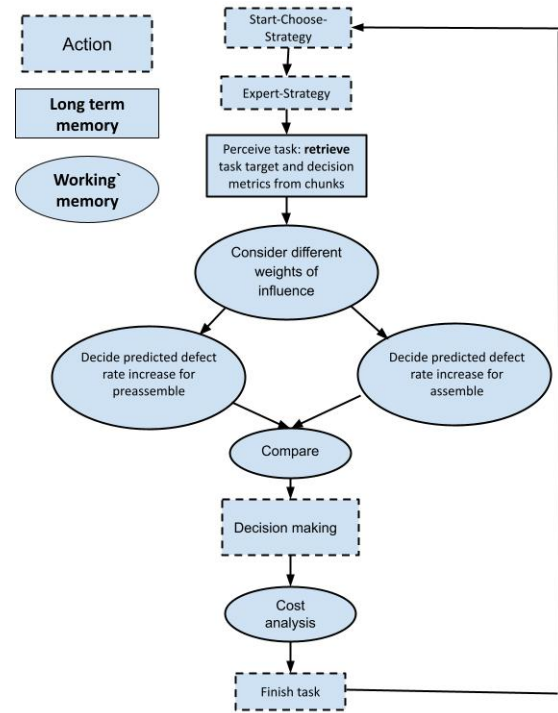


Fig. 3. Production rules control structure for expert decision making and their use of the ACT-R Goal and Imaginal buffers

strategy used before looping back to the next decision-making round.

Level of Expertise Mechanism

The model can learn while performing tasks through a mechanism leading to varying levels of expertise, as shown in Figure 4. The model mimics human decision-making behavior through differentiating knowledge representations. **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 represent a portion of the intuition used by novice decision-makers, while the red circles contains VSM domain knowledge used by intermediate 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 their production rule sets. Finally, the goal buffer uses the ‘goal

focus' command to manipulate the different phases of the task.

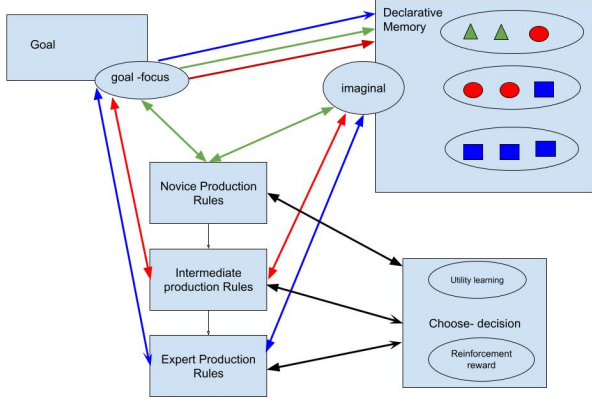


Fig. 4. Level of expertise mechanism in VSM-ACT-R

The model also simulates the learning progress through the **Decision-Choice Control**, which manages errors, learning, and memory via utility learning and reinforced rewards. Novice decision-making productions start with a utility base and include a noise parameter. Each round of decisions receives rewards or penalties, and the utility of associated production rules updates with the adjustment of memory retention, which depends on the time passed since the rule last fired.

Foster Metacognition to Support Learning

With the aim of making the model assess the effectiveness of decisions while learning — akin to human metacognition, self-assessing and self-correcting in response to self-assessment [57] — we consequently developed a dynamic reward function that rewards actions after self-evaluating the chosen strategy.

VSM-ACT-R uses the Temporal Difference (TD) algorithm from reinforcement learning [76] as expressed in Eqn. 1. Each production rule in the ACT-R model has a utility—a value or strength—associated with it, which is updated using the TD algorithm:

$$\text{Eqn. 1 : } U_i(n) = U_i(n-1) + \alpha [R_i(n) - U_i(n-1)]$$

where $U_i(n)$ represents the value or utility of some item i (i.e., a production) after its n -th occurrence, and $R_i(n)$ represents the reward received on the n -th occurrence. The parameter α ($0 < \alpha < 1$) controls the learning rate. If multiple productions compete with expected utility values U_j , the probability of selecting

production i is given by Eqn.2:

$$\text{Eqn. 2 : } \text{Probability}(i) = \frac{e^{U_i/\sqrt{2s}}}{\sum_j e^{U_j/\sqrt{2s}}},$$

where the summation over j is over all the productions that currently have their conditions satisfied; and s is a noise parameter.

The utilities of production are learned as the model runs, based on the rewards or penalty that are received. Where we designed the reward function as $R(s, f(x))$ that calculates the reward at the end of each decision-making round. This function takes two parameters: S , representing the strategy used, and $f(x)$, which results from headcount cost analysis, leading to either a weighted reward or a penalty. For example, in one decision round, a penalty of -2 is computed due to the use of a novice strategy coupled with inefficient headcount cost analysis. Factoring in the memory retention effect after a 0.05 seconds step, the calculation using the TD algorithm modifies the impact of the decision on the utility of the next production as:

$$U_i(n+1) = U_i(n) + \alpha [-2 - 0.05 - U_i(n)].$$

This will then sequentially update the utility of the chain of productions for the chosen strategy. We find that when the model encounters certain types of problems where both novice and expert strategies result in similar efficiencies in cost assessment. In these cases, the model is prone to staying with the novice strategy and exhibits a more gradual learning curve, similar to the tendency for people facing bounded rationality in decision-making [25, 31], where they are likely to select the less effortful option when faced with multiple choices that produce very similar outcomes.

VSM-ACT-R model evaluation

To answer the question of whether VSM-ACT-R decisions demonstrate learning progression, and capture individual differences, this study first uses descriptive statistics and linear regression to show the average progression of decision types across trials. It then use a mixed linear model to assess and illustrate the effects of trials on decision types across ACT-R model personas, with repeated measures of trials, and random effects to account for individual differences. Last but not least, it uses ordered logistic regression to analyze and understand the relationship between the number

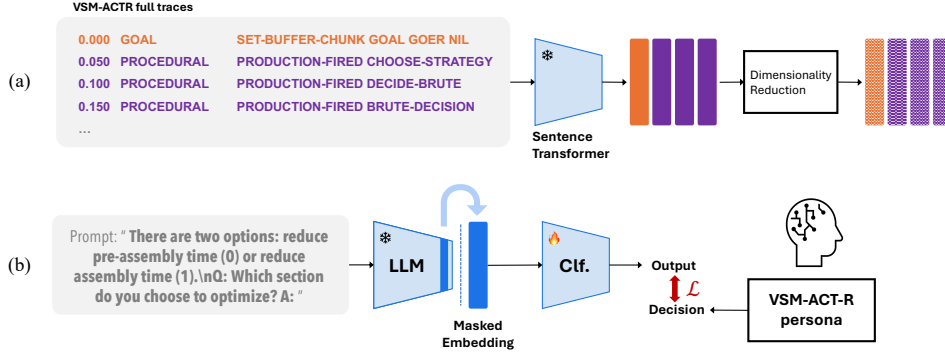


Fig. 5. (a) Obtaining decision representations from VSM-ACT-R. (b) LLM feature extraction for behavior prediction.

of trials and an ordinal dependent variable of learning progress from novice to expert.

We ran the VSM-ACTR model 2,012 times to understand its behavior [67]. Each time, we asked it to run 15-16 trials until the model achieved stable expert behavior. We collected data with decision types encoded as 0, 1, and 2 for novice, intermediate, and expert strategies.

Fig. 6 shows a significant positive impact of trial exposure on decision-making progression, evidenced by a linear coefficient of 0.086 ($P < 0.05$). Furthermore, the standard deviation starts relatively low but quickly increases, peaking around the third trial. This could reflect a diverging approach to decision-making as VSM-ACTR personas experiment with different strategies. The standard deviation gradually decreases thereafter, stabilizing between 0.5 and 0.75, which points to a convergence in decision-making strategies among personas. 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, suggesting that the trials themselves predominantly explain the variability in decision type, while the individual differences exist. 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. These findings validate that the repeated reinforcement decisions from VSM-ACTR demonstrate human-like learning progression and capture individual differences.



Fig. 6. Trend of decision types over trials, blue line is average decision types, red line is variance

The Knowledge Transfer Framework: LLM-ACTR

With the validated model in hand, we then explain the LLM-ACTR framework, beginning with its cognitive knowledge input, followed by its knowledge transfer mechanism.

Cognitive Decision-Making Knowledge

This study curated VSM-ACTR decision-making knowledge through VSM-ACTR's traces, which capture the reasoning steps in real time using a concurrent protocol. These traces log the cognitive operations executed by the modules at each decision point. The traces exhibit metacognition, which involves awareness and understanding of one's own decision-making processes. This is represented through model traces that demonstrate the use of the imaginal buffer for accessing working memory, procedural memory matching and firing, and the self-assessment of strategy effectiveness. Traces also exhibit executive function [27], which involves the evolution of decision-making results across trials and shows how decisions adapt through learning and experience.

Table 1: VSM-ACTR decision-making trace that highlights goal initiation, strategy selection, decision evaluation, utility update, and learning.

001	0.000	GOAL SET-BUFFER-CHUNK GOAL GOER NIL
002	0.050	PROCEDURAL PRODUCTION-FIRED CHOOSE-STRATEGY
003	0.100	PROCEDURAL PRODUCTION-FIRED DECIDE-BRUTE
004	0.150	PROCEDURAL PRODUCTION-FIRED BRUTE-DECISION
005		assembly is always a good place to reduce time!
006	0.200	PROCEDURAL PRODUCTION-FIRED HEADCOUNT
007	-0.01999998	
008	0.250	PROCEDURAL PRODUCTION-FIRED STOP
009		this is the end of one decision making
010		Utility updates with Reward = -2.0 alpha = 0.2
011		Updating utility of production CHOOSE-STRATEGY
012	U(n-1) = 0.0 R(n) = -2.25 [-2.0 - 0.25 seconds since selection]	
013	U(n) = -0.45000002	
014		Updating utility of production DECIDE-BRUTE
015	U(n-1) = 3.0 R(n) = -2.2 [-2.0 - 0.2 seconds since selection]	
016	U(n) = 1.96	
...		
026	0.300	PROCEDURAL PRODUCTION-FIRED CHOOSE-STRATEGY
027	0.350	PROCEDURAL PRODUCTION-FIRED DECIDE-INTERMEDIATE
...		
056	0.800	PROCEDURAL PRODUCTION-FIRED CHOOSE-STRATEGY
057	0.850	PROCEDURAL PRODUCTION-FIRED EXPERT-STRATEGY
...		
084		Updating utility of production CHOOSE-STRATEGY
085	U(n-1) = -0.46 R(n) = 4.65 [6.0 - 1.35 seconds since selection]	
086	U(n) = 0.56200004	
087		Updating utility of production EXPERT-STRATEGY
088	U(n-1) = 0.0 R(n) = 4.7 [6.0 - 1.3 seconds since selection]	
089	U(n) = 0.94	

As shown in Table 1, the model begins by establishing the goal (line 1) and then proceeds with a novice strategy (line 3, BRUTE). For the production rules associated with each strategy, the utility of each production rule is updated based on the received reward and the time since the last selection. For instance, the reward computation based on cost analysis (line 6) for the BRUTE choice results in a reward of -2 (line 10). Consequently, the utility of the NAIVE-CHOICE rule, impacted by a penalty of -2.25 for the time passed since the last selection, decreases from 3 to 1.96 (lines 14-16). As the utility of naive strategies declines, the probability of triggering the Intermediate Strategy (lines 26-27) and the EXPERT Strategy (lines 87-89) increases.

Learning an Embedding Space of Decision Traces

The next step is to convert the traces into vectors that LLMs can process. To retain executive function processes, we log decision results and strategy traces, which are then numerically encoded. For instance, 0' represents a decision for reduced time in the preassembly section, and 1' for assembly. Encoded data are subsequently fed into the neural network as single vectors.

To retain both executive function and metacognition processes, this study employs a semantic extraction and dimensionality reduction approach. This approach aims to transform a vast number of cognitive reasoning stamps into a vector format that balances information retention with computational efficiency. Traces for each task are processed through a sentence transformer to obtain semantic embeddings for each timestamp. 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. These embeddings are then reduced to N dimensions using Principal Component Analysis (PCA) [1] (see Figure 5a). The learned embeddings can then be concatenated into a one-dimensional vector that serves as a content vector. This content vector could then be used to elicit meaningful cognitive decision-making behavior perturbations in LLMs. For example, the preliminary experiment explores the transfer of both metacognitive and executive function processes into LLMs by adding the cognitive content vector to the forward pass of LLM next token prediction to elicit meaningful behavioral perturbations.

Transfer of Learning

LLM-ACTR (see Fig. 1(4) LLM-ACTR framework) begins by (a) parsing consistent template prompts that reflect the decision making task into an open-source LLM, mapping the task for the cognitive model; (b) using the LLM as the base model to access the last hidden layer and obtain masked embeddings; (c) constructing a classification layer with softmax activation on top of the base model; (d) 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.

Fine-tuning, which involves optimizing model weights for a specific task, has been widely applied in the transfer of learning [29]. Aiming at transferring human-like decisions with learning, the targets are the encoded vectors that represent executive function processes of each VSM-ACTR persona. The transfer of learning has been reformulated into a classification fine-tuning task, where the final layer of contextualized embeddings—capturing the in-context meaning of tokens by recombining them with other tokens' embeddings—is used as features. These selected contextualized embeddings provide the richest semantic information while balancing minimal information loss and reduced computational costs for fine-tuning.

1 Additionally, Low-Rank Adaptation (LoRa) was em-
 2 ployed for its computational efficiency [33]. The cur-
 3 rent LLM-ACTR framework can also be extended to
 4 transfer other cognitive processes such as metacogni-
 5 tion, as demonstrated in the following preliminary ex-
 6 periments section.

9 Experiments

11 *Use Semantic Mapping to Evaluate Cognitive* 12 *Decision Making Traces Vector*

13 To answer RQ1 regarding the properties of a neural
 14 network representation of the decision-making process
 15 in CAs, we conducted a semantic mapping analysis of
 16 the first two principal components of the learned em-
 17 beddings of each trace. The goal is to explore how
 18 the neural network has the potential to learn guided
 19 perception, memory, goal-setting, and actions — key
 20 components of cognitive decision-making — in an em-
 21 bedding space. We then used MANOVA analysis to ex-
 22 amine how the learned embeddings correspond to the
 23 semantic of ACT-R’s components, including procedu-
 24 ral memory, imaginal memory, goal knowledge, utility
 25 updating, and decision-making actions.

27 *Feature Extraction for Behavior Prediction*

28 To answer RQ2: What level of complexity in behav-
 29 ior representation can LLMs effectively capture? This
 30 study adopted the similar method of LLMs’ feature
 31 extraction for behavior prediction [36]. We created
 32 datasets consisting of LLMs’ last contextual embed-
 33 dings as features and the corresponding different lev-
 34 els of VSM-ACTR decisions as targets. We obtained
 35 embeddings by passing prompts that included all the
 36 information that VSM-ACTR had access to on a given
 37 trial and then extracting the hidden activations of the
 38 final layer, as shown in Figure 5b.

39 The first dataset used targets as VSM-ACTR deci-
 40 sions, where ‘0’ indicates preassembly and ‘1’ indi-
 41 cates assembly. The second dataset’s prompt template
 42 added an explanation of the strategy adopted by VSM-
 43 ACTR (see Appendix: LLM System Prompt Tem-
 44 plates) and used compound targets comprising both the
 45 decisions and the strategies reflecting the learning tra-
 46 jectory (novice, intermediate, and expert). The targets
 47 were encoded as follows: 0, 1, and 2 for preassembly
 48 choices using novice, intermediate, and expert strate-
 49 gies, respectively, and 3, 4, and 5 for assembly choices

1 following the same pattern. With these two datasets,
 2 we fitted a regularized logistic regression model using
 3 10-fold cross-validation for the first dataset and multi-
 4 nomial regression using 10-fold cross-validation with
 5 L2 regularization for the second. Model performance
 6 was assessed by measuring the goodness of fit through
 7 negative log-likelihood (NLL) and the predictive accu-
 8 racy of hold-out data.

9 *Knowledge Transfer*

10 To answer RQ3: whether LLMs can be informed with
 11 knowledge about the reasoning processes of cogni-
 12 tive architectures, we use a case study to examine
 13 whether Cognitive LLMs offer better representa-
 14 tions of human decision-making behaviors on a novel
 15 Design for Manufacturing problem, compared to an
 16 LLM-only model that employs chain-of-thought rea-
 17 soning strategies.

20 *Base Model and Data*

21 The case study uses the LLaMa-2 13B [79] model as
 22 the base model because it demonstrated effectiveness
 23 and efficiency in NLP tasks [35]. As a state-of-the-
 24 art LLM, LLaMa has been trained on trillions of to-
 25 kens from publicly available datasets. Unlike other
 26 transformer-based models such as the GPT family,
 27 which can only be accessed at the user’s end, LLaMa’s
 28 architecture, including its pre-trained weights, is fully
 29 accessible. Furthermore, evidence that its internal rep-
 30 resentations can be trained to become more aligned
 31 with human neural activity has been presented [12].

32 To determine the target size that can effectively per-
 33 form the fine-tuning task while balancing efficacy and
 34 resource limitations, we referred to [45], who showed
 35 evidence that LLaMa-2 13B would maintain competi-
 36 tive performance in resource-limited text classification
 37 with datasets of nearly 1,000 rows per class. Based
 38 on this, we created a dataset that contains the 2,012
 39 decision-making trials, obtained by running the devel-
 40 oped VSM-ACTR model across 32 problem sets; each
 41 ACT-R persona was run for 15-16 trials until stable ex-
 42 pert behavior was achieved.

44 *Experiment Metrics*

45 The fine-tuning process employs cross-entropy as the
 46 loss function and uses Adam optimization. Training in-
 47 volves a train-test split of 0.2 and a batch size of 5
 48 for both training and validation phases. The learning
 49 rate was set to 1e-5, with training spanning across 10
 50 epochs. To ensure regularization and prevent overfit-
 51 ting, weight decay of 0.01, a dropout rate of 0.5 were

1 applied, and gradient accumulation was set to 2. Last
2 but not least, gradient clipping was employed to main-
3 tain a maximum gradient norm of 1.0 for gradient ex-
4 plosion control.

5 Baseline Models

6 To assess the model’s ability to make human-like de-
7 cisions, we first split the data into train and validation
8 sets to reserve a set of unseen problems. We then com-
9 pared the predictive negative log-likelihood (NLL),
10 a measure of goodness-of-fit, of Cognitive LLMs
11 in predicting VSM-ACTR’s decisions on the unseen
12 problems, against a pre-trained LLaMa and a random
13 guess model.

14 A random choice model serves as the basic form of
15 control condition to distinguish the effects of treatment
16 from chance [26]. This approach allows assessing the
17 extent to which decisions are influenced by knowledge
18 versus being purely stochastic. On the other hand, us-
19 ing LLaMa without fine-tuning as a baseline provides
20 a reference point to measure the impact of knowledge
21 transfer on the model’s performance.

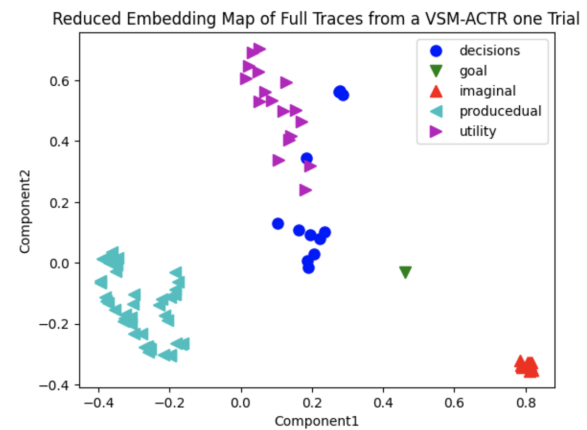
25 Results

27 Finding Useful Cognitive Decision Making 28 Embeddings

29 The approach of distilling executive function pro-
30 cesses captures the evolution of decision-making re-
31 sults across trials and illustrates how decisions adapt
32 through learning and experience, all represented as a
33 sequential single vector. This approach is easy to use
34 for downstream tasks but retains only partial knowl-
35 edge of cognitive decision-making

36 In addition, Figure 7 displays the reduced embeddings
37 of both metacognitive and executive function pro-
38 cesses corresponding to the semantic mapping of ACT-
39 R’s components. The MANOVA analysis was con-
40 ducted to assess the overall effect of the independent
41 variables, including label categories or ACT-R compo-
42 nents, on the combined dependent variables—components
43 of reduced embeddings. This analysis reveals a signif-
44 icant relationship with the semantic mapping of ACT-
45 R’s components. For instance, the Wilks’ lambda value
46 (0.0004) suggests that the label or ACT-R component
47 categories explain nearly all the variance in the depen-
48 dent variables, indicative of a strong group effect. The
49 statistical tests applied—Wilks’ lambda, Pillai’s trace,

1 Hotelling-Lawley trace, and Roy’s greatest root—all
2 demonstrate strong significance, as evidenced by p-
3 values less than 0.05 across all tests. It shows that the
4 semantics of symbolic and subsymbolic representa-
5 tions of cognitive models can be learned using a neural
6 network, and the principal components retained suc-
7 cessfully capture the essential variance related to these
8 cognitive processes, providing a way to preserve cog-
9 nitive decision-making knowledge in a compact em-
10 bedding space.



12 Fig. 7. Reduced embedding map to full traces from VSM-ACTR one
13 trail

29 Assessing Behavior Complexity Captured by the 30 Innate LLM

31 Table 2 shows that LLM-ACTR captures a single
32 facet of decision-making, achieving an average accu-
33 racy of 0.64 across 10 validation folds in the holdout
34 task. When decision-making targets involve multiple
35 facets—encompassing both choices and strategies that
36 shape the learning trajectory—the accuracy decreases
37 to 0.42. In addition, the NLL reveals greater predic-
38 tive uncertainty for multifaceted decision-making pro-
39 cesses, as evidenced by a significantly higher NLL
40 of 1.18 compared to 0.65 in single-facet scenarios.
41 The results show that prompt embeddings generated
42 through feature extraction capture the overall structure
43 of learning. However, they struggle to capture complex
44 decision-making rationales.

47 Learning Cognitive Decision-Making through 48 LLM-ACTR

49 We first report training and validation losses, across
50 10 epochs, to reveal the fine-tuned model’s learning
51

Table 2: 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 3: 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

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.

We then report the comparison of the Cognitive LLMs with the baseline models on goodness of fit using negative log likelihood (NLL) and accuracy score for hold-out data. The Cognitive LLMs demonstrates significantly better performance across all metrics compared to the LLaMa-only model, highlighting its effectiveness in decision-making tasks involving reinforced learning. Additionally, the LLaMa-only model performs worse than the chance-level model. We believe this underscores the necessity of fine-tuning pre-trained language models like LLaMa to adapt them to human-like decision-making patterns.

Preliminary Experimental Results on Extending LLM-ACTR

Following results for RQ1 that the semantics of symbolic and subsymbolic representations of cognitive models can be learned using a neural network, we conducted a preliminary experiment to extend LLM-ACTR to transfer holistic cognitive processes.

After retaining a randomly-chosen 240 full cognitive reasoning traces from the VSM-ACTR model, we processed both executive function and metacognition pro-

cesses using a semantic extraction and dimension reduction approach (see Figure 5a). The resulting embeddings were concatenated into 240 one-dimensional tensors. We then addressed the issue of ragged tensors due to the individual difference by padding, then calculated the standardized mean values of these tensors to serve as a content vector.

The preliminary experiment extends LLM-ACTR with the content vector into training. The content vector is injected into one of the hidden layers during a forward pass to introduce differentiated activations. Using the modified LLM as the base model, it accesses the last contextualized embedding and obtains the masked embedding. A classification layer with softmax activation is constructed on top to form the decision-making layer. Using targets of ACTR model decisions, the Cognitive LLM is fine-tuned for the classification task in decision-making using LoRA (see Fig. 8). We switched to a smaller size of LLaMa 7b for the experiment to strike a balance between the computational costs of back propagation when modifying the model’s hidden layers and the overall efficacy of the base model.

The LLaMa model with the modified hidden layer is fine-tuned with 2,012 data points for the binary classification task. The content vectors are set to be trainable. To assess the model’s ability to make human-like decisions, we first split the data into train and validation sets to reserve a set of unseen problems. We then compared the predictive NLL of Cognitive LLM in predicting VSM-ACTR’s decisions on the unseen problems, against LLaMa fine-tuned without content vectors.

The results (Fig. 9) show that the addition of the vector representation of VSM-ACTR’s holistic traces during fine-tuning resulted in a slightly decreased mean and reduced variance of NLL across 10 epochs, demonstrating better model fitting and stability compared to fine-tuning only. It indicates that allowing the model to integrate and learn from the cognitive vector during training potentially leads to more nuanced and human-like decision-making capabilities, as captured by the cognitive features encoded in the vector. However, the influence of the cognitive content vector is limited and warrants further investigation, partly because the stochastic simulation of the VSM-ACTR produces decision-making vectors of various lengths. This study addresses ragged tensors by padding, but this approach potentially dilutes or changes the semantics of each

vector. To improve the impact of the cognitive vector, additional techniques such as vector optimization will be needed.

Discussion and Conclusion

Main Insights/Takeaways This paper starts to show how to enable LLMs to replicate cognitive decision-making in CAs via a data-driven approach. We introduce *Cognitive-LLMs*, a novel neuro-symbolic architecture designed to enhance human-like decision-making by integrating the CAs' cognitive processes with LLMs. (1) It introduces *VSM-ACTR*, a human-like cognitive model for manufacturing solutions, developed using the *ACT-R CA*. The model exhibits reinforcement learning in decision-making and can be used to model metacognitive processes to reflect on and evaluate the effectiveness of the actions. (2) It then examines latent representations of CAs through neural networks. The findings show that distilling the executive function process preserves high-level symbolic knowledge but only partially capturing decision-making involves learning. A holistic semantic preservation approach, covering both executive function and metacognitive processes, retains symbolic and sub-symbolic semantics in a low-dimensional space. However, challenges with ragged tensors derived from individual differences in downstream tasks require further optimization. (3) We then collected domain knowledge as the executive function process and used the knowledge as labeled targets in a feature extraction for behavior prediction task to investigate the LLMs' innate capabilities in capturing the complexity of behavioral representations. The results show that prompt embeddings generated through feature extraction capture the overall structure of learning. However, they struggle to capture complex decision-making rationales.

Further more, (4) This study presents a developing framework *LLM-ACTR* for knowledge transfer from cognitive models to LLMs, rooted in the mechanism of LLMs' next-token prediction and the knowledge representation of cognitive models. This includes methods such as using the cognitive models' decisions for fine-tuning [29], and integrating a cognitive decision-making vector into hidden layer to elicit meaningful behavior perpetuation [62]. (5) It advances previous efforts on human-like LLMs alignment using data from large-scale cognitive psychology experiments involving human subjects [11, 19]. It reduces the cost of

data collection by using synthetic data from cognitive models. The synthetic data present real-time cognitive reasoning with tasks, including metacognition, which is hard to quantify in human subjects [23]. (6) The case study of *Cognitive LLMs* in manufacturing decision-making demonstrates that *Cognitive LLMs* achieves better fitting of human-like decisions on unseen problems compared to a pre-trained model in the DFM task. Thus, it is possible to transfer decision-making knowledge from CAs to LLMs.

This development opens up new research directions for equipping LLMs with the necessary knowledge to computationally model and replicate the internal mechanisms of human cognitive decision-making [59, 60]. It also complements ongoing work showing that LLMs could possibly be transformed into cognitive models through knowledge transfer, e.g., [12, 18, 19]. For example, [13] shows that through fine-tuning, LLMs' internal representations become more aligned with human behaviors.

Limitations and Future Work One limitation also stems from the novelty of this study. How closely can we claim that cognitive model personas replicate human behaviors? Currently, our focus is on tuning the model to align with general patterns of learning and error-making; however, *VSM-ACTR* still requires more granular human data for cognitive fine-tuning. The closer the *VSM-ACTR* model aligns with human behavior, the more accurately it can represent human decision-making processes and explain human behavior.

However, the more meaningful questions arise from considering the landscape of enabling machine cognitive reasoning. We must ask ourselves what we can learn about cognitive decision-making when we infuse knowledge from CAs into LLMs. For now, our insights are limited to the observation that knowledge from cognitive models can be preserved in an embedding space and could be learned by LLMs, and that embeddings from large language models can be trained to predict human-like decisions. While this is interesting in its own right, it certainly is not the end of the story. Looking beyond the current work, transitioning from transferring cognitive models' human-like decisions to LLMs, to guided perception, memory, goal-setting, and actions, will provide the opportunity to apply a wide range of explainability techniques to LLMs' cognitive decision-making.

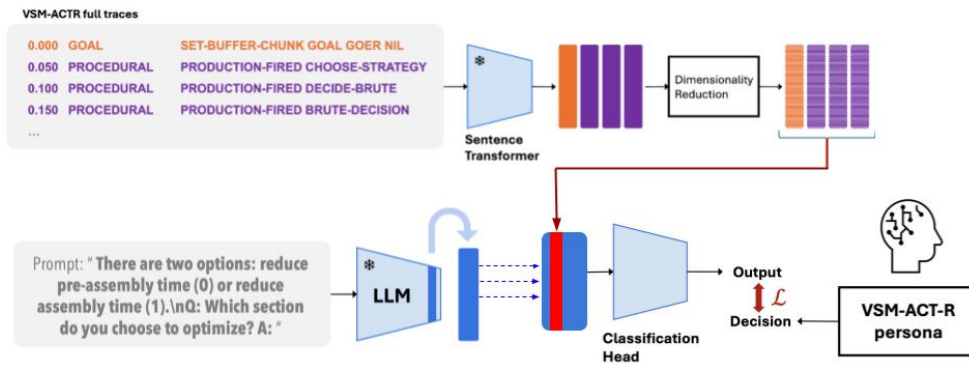


Fig. 8. Infusing holistic VSM-ACTR traces as content vectors through fine-tuning.

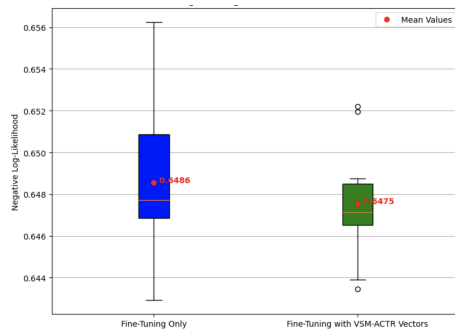


Fig. 9. Comparison of NLL across 10 epochs for fine-tuning only and fine-tuning with cognitive content vectors

One application of this further work can be used to address a common limitation in machine learning innovations — cross domain generalization, e.g., [4, 92]. Cognitive LLMs can currently only generalize to unseen problems within an applicable domain, constrained by fixed decision candidates and unknown decision metric values. In applying Cognitive LLMs to evolving manufacturing problems that may incorporate an increasing number of decision candidates and associated metrics, it becomes critical to solve out-of-domain problems [83]. This will require *LLM-ACTR* to advance in transferring guided perception, memory, and goal-setting to LLMs. As Zhu and Simmons [93] found, training the LLM with the rules of guided perception in cognitive models can help generalize robotics problem-solving to out-of-distribution tasks.

Credit Author Statement

Siyu Wu: Conceptualization, Methodology, Software, Experiments, Writing - Original Draft, Writing - Review & Editing. Alessandro Oltramari: Conceptu-

alization, Funding Acquisition, Methodology, Software, Writing - Review & Editing. Jonathan Francis: Methodology, Experiments, Writing- Review & Editing. C. Lee Giles: Conceptualization, Writing - Review & Editing. Frank E. Ritter: Writing - Review & Editing.

References

- [1] H. Abdi and L.J. Williams, Principal Component Analysis, *Wiley Interdisciplinary Reviews: Computational Statistics* **2**(4) (2010), 433–459.
- [2] S. Abdurahman, H. Vu, W. Zou, L. Ungar and S. Bhatia, A deep learning approach to personality assessment: Generalizing across items and expanding the reach of survey-based research, *Journal of Personality and Social Psychology* (2023), Advance online publication. doi:10.1037/pspp0000480.
- [3] A. Aka and S. Bhatia, Machine learning models for predicting, understanding, and influencing health perception, *Journal of the Association for Consumer Research* **7**(2) (2022), 142–153. doi:10.1086/718456.
- [4] M. Akrouf, A. Feriani, F. Bellili, A. Mezghani and E. Hosain, Domain generalization in machine learning models for wireless communications: Concepts, state-of-the-art, and open issues, *IEEE Communications Surveys & Tutorials* **25** (2023), 3014–3037.
- [5] J.R. Anderson, *How can the human mind occur in the physical universe?*, Oxford University Press, 2009.
- [6] J.R. Anderson, S. Betts, D. Bothell, R. Hope and C. Lebiere, Learning rapid and precise skills, *Psychological Review* **126** (2019), 727–760.
- [7] J. Andreas, Language models as agent models, in: *Findings of the Association for Computational Linguistics: EMNLP 2022*, 2022, pp. 5769–5779.
- [8] G. Bao, H. Zhang, L. Yang, C. Wang and Y. Zhang, LLMs with chain-of-thought are non-causal reasoners, *arXiv preprint arXiv:2402.16048* (2024).
- [9] 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*, 2003, pp. 64–72.

- [10] S. Bhattamishra, A. Patel, P. Blunsom and V. Kanade, Understanding in-context learning in transformers and LLMs by learning to learn discrete functions, *arXiv preprint arXiv:2310.03016* (2023).
- [11] M. Binz and E. Schulz, Using cognitive psychology to understand GPT-3, *Proceedings of the National Academy of Sciences* **120**(6) (2023), e2218523120.
- [12] 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.
- [13] M. Binz, E. Akata, M. Bethge, F. Brändle, F. Callaway, J. Coda-Forno, P. Dayan, C. Demircan, M.K. Eckstein, N. Éltető, T.L. Griffiths, S. Haridi, A.K. Jagadish, L. Ji-An, A. Kipnis, S. Kumar, T. Ludwig, M. Mathony, M. Mattar, A. Modirshanechi, S.S. Nath, J.C. Peterson, M. Rmus, E.M. Russek, T. Saanum, N. Scharfenberg, J.A. Schubert, L.M.S. Buschoff, N. Singhi, X. Sui, M. Thalmann, F. Theis, V. Truong, V. Uandarao, K. Voudouris, R. Wilson, K. Witte, S. Wu, D. Wulff, H. Xiong and E. Schulz, Centaur: a foundation model of human cognition, *arXiv:2410.20268* (2024).
- [14] D. Bothell, *ACT-R 7 Reference Manual*, 2017, Available at <http://act-r.psy.cmu.edu/wordpress/wp-content/themes/ACT-R/actr7/reference-manual.pdf>.
- [15] T.B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal et al., Language models are few-shot learners, in: *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 33, 2020, pp. 1877–1901.
- [16] N. Chakraborty, M. Ornik and K. Driggs-Campbell, Hallucination detection in foundation models for decision-making: A flexible definition and review of the state of the art, *arXiv* (2024).
- [17] H. Chen, S. Liu, L. Pang, X. Wanyan and Y. Fang, Developing an Improved ACT-R Model for Pilot Situation Awareness Measurement, *IEEE Access* **9** (2021), 122113–122124.
- [18] J. Coda-Forno, M. Binz, J.X. Wang and E. Schulz, CogBench: A large language model walks into a psychology Lab, in: *Accepted to the International Conference on Learning Representations (ICLR)*, 2024.
- [19] J. Coda-Forno, M. Binz, Z. Akata, M. Botvinick, J. Wang and E. Schulz, Meta-in-context learning in large language models, in: *Advances in Neural Information Processing Systems*, Vol. 36, 2024.
- [20] Q. Dong, L. Li, D. Dai, C. Zheng, Z. Wu, B. Chang, X. Sun, J. Xu and Z. Sui, A survey on in-context learning, *arXiv preprint arXiv:2301.00234* (2022).
- [21] M. Dorobantu, Human-level, but non-humanlike: Artificial intelligence and a multi-level relational interpretation of the imago Dei, *Philosophy, Theology and the Sciences* **8**(1) (2021), 81–107.
- [22] V.G. dos Santos, G.L. Santos, T. Lynn and B. Benatallah, Identifying citizen-related issues from social media using LLM-based data augmentation, in: *International Conference on Advanced Information Systems Engineering*, Springer Nature Switzerland, Cham, 2024, pp. 531–546.
- [23] S.M. Fleming and H.C. Lau, How to measure metacognition, *Frontiers in Human Neuroscience* **8** (2014), 443.
- [24] J. Francis, N. Kitamura, F. Labelle, X. Lu, I. Navarro and J. Oh, Core challenges in embodied vision-language planning, *Journal of Artificial Intelligence Research* **74** (2022), 459–515.
- [25] W.-T. Fu and W.D. Gray, Resolving the paradox of the active user: Stable suboptimal performance in interactive tasks, *Cognitive Science* **28** (2004), 901–935.
- [26] J. Gaab, J. Kossowsky, U. Ehlerer and C. Locher, Effects and components of placebos with a psychological treatment rationale—Three randomized-controlled studies, *Scientific Reports* **9**(1) (2019), 1421.
- [27] S.J. Gilbert and P.W. Burgess, Executive function, *Current Biology* **18**(3) (2008), R110–R114.
- [28] C. Gonzalez, J.F. Lerch and C. Lebiere, Instance-based learning in dynamic decision making, *Cognitive Science* **27** (2003), 591–635. doi:10.1207/s15516709cog2704_2.
- [29] Y. Guo, H. Shi, A. Kumar, K. Grauman, T. Rosing and R. Feris, Spottune: Transfer learning through adaptive fine-tuning, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4805–4814.
- [30] T. Hagendorff, S. Fabi and M. Kosinski, Human-like intuitive behavior and reasoning biases emerged in large language models but disappeared in ChatGPT, *Nature Computational Science* **3**(10) (2023), 833–838.
- [31] R. Hastie and R.M. Dawes, *Rational choice in an uncertain world: The psychology of judgment and decision making*, Sage, 2010.
- [32] E. Hozdić, Smart Factory for Industry 4.0: A review, *International Journal of Modern Manufacturing Technologies* **7**(1) (2015), 28–35.
- [33] E.J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang and W. Chen, LoRA: Low-rank adaptation of large language models, in: *International Conference on Learning Representations*, 2022. <https://openreview.net/forum?id=nZeVKeeFYf9>.
- [34] Y. Hu, Q. Xie, V. Jain, J. Francis, J. Patrikar, N. Keetha, S. Kim, Y. Xie, T. Zhang, Z. Zhao et al., Toward general-purpose robots via foundation models: A survey and meta-analysis, *arXiv preprint arXiv:2312.08782* (2023).
- [35] D. Huang, Z. Hu and Z. Wang, Performance analysis of Llama 2 among other LLMs, in: *2024 IEEE Conference on Artificial Intelligence (CAI)*, 2024, pp. 555–559.
- [36] Z. Hussain, M. Binz, R. Mata and D.U. Wulff, A tutorial on open-source large language models for behavioral science, *PsyArXiv preprint* (2023).
- [37] H. Joshi and V. Ustun, Augmenting cognitive architectures with large language models, in: *Proceedings of the AAAI Symposium Series*, Vol. 2, 2024. doi:10.1609/aaais.v2i1.27689.
- [38] M. Kang, Team-Soar: A computational model for multilevel decision making, *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* **31**(6) (2001), 708–714. doi:10.1109/3468.983407.
- [39] J.R. Kirk, R.E. Wray and J.E. Laird, Exploiting language models as a source of knowledge for cognitive agents, in: *Proceedings of the AAAI Symposium Series*, Vol. 2, 2023, pp. 286–294.
- [40] J.R. Kirk, W. Robert, P. Lindes and J.E. Laird, Improving knowledge extraction from LLMs for robotic task learning through agent analysis, 2023.
- [41] J.R. Kirk, R.E. Wray, P. Lindes and J.E. Laird, Improving knowledge extraction from LLMs for task learning through agent analysis, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38, 2024, pp. 18390–18398.
- [42] T. Kojima, S.S. Gu, M. Reid, Y. Matsuo and Y. Iwasawa, Large language models are zero-shot reasoners, 2022, <https://api.semanticscholar.org/CorpusID:249017743>.

- [43] I. Kotseruba and J.K. Tsotsos, A review of 40 years of cognitive architecture research: Focus on perception, attention, learning and applications, *AI Review* **53** (2020), 17–94.
- [44] I. Kotseruba and J.K. Tsotsos, *The computational evolution of cognitive architectures*, Oxford University Press, in press.
- [45] A. Kumar, R. Sharma and P. 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) (2024), 14219–14224.
- [46] J.E. Laird, *The Soar cognitive architecture*, MIT Press, 2012.
- [47] J.E. Laird, C. Lebiere and P.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) (2017), 13–26.
- [48] J.E. Laird, An analysis and comparison of ACT-R and Soar, in: *Proceedings of the Ninth Annual Conference on Advances in Cognitive Systems*, 2021.
- [49] M. Leyer and S. Schneider, Decision augmentation and automation with artificial intelligence: Threat or opportunity for managers?, *Business Horizons* **64**(5) (2021), 711–724.
- [50] H. Liang, X. Sun, Y. Sun and Y. Gao, Text feature extraction based on deep learning: A review, *EURASIP Journal on Wireless Communications and Networking* (2017), 1–12.
- [51] R. Liu, J. Geng, J.C. Peterson, I. Sucholutsky and T.L. Griffiths, Large language models assume people are more rational than we really are, *arXiv cs.CL* (2024). <https://doi.org/10.48550/arXiv.2406.17055>.
- [52] L. Makatura, M. Foshey, B. Wang, F. Hähnlein, P. Ma, B. Deng, M. Tjandrasuwita, A. Spielberg, C.E. Owens, P.Y. Chen, A. Zhao, A. Zhu, W.J. Norton, E. Gu, J. Jacob, Y. Li, A. Schulz and W. Matusik, How can large language models help humans in design and manufacturing? Part 2: synthesizing an end-to-end LLM-enabled design and manufacturing workflow, *Harvard Data Science Review* (2024).
- [53] T. Manos, Value Stream Mapping-An introduction, *Quality Progress* **39**(6) (2006), 64.
- [54] J.N. Marewski and K. Mehlhorn, Using the ACT-R architecture to specify 39 quantitative process models of decision making, *Judgment and Decision Making* **6**(6) (2011), 439–519.
- [55] 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*, Carnegie Mellon University/University of Pittsburgh, Pittsburgh, PA, 2004.
- [56] J. McCarthy, From here to human-level AI, *Artificial Intelligence* **171**(18) (2007), 1174–1182.
- [57] T.O. Nelson and L. Narens, *Why investigate metacognition?*, in: *Metacognition: Knowing about knowing*, The MIT Press, 1994, pp. 1–25.
- [58] A. Newell, *Unified Theories of Cognition*, Harvard University Press, 1994.
- [59] A. Oltramari, Enabling high-level machine reasoning with cognitive neuro-symbolic systems, in: *Proceedings of the AAAI Symposium Series*, Vol. 2, 2023, pp. 360–368.
- [60] A. Oltramari, J. Francis, F. Ilievski, K. Ma and R. Mirzaee, Generalizable neuro-symbolic systems for commonsense question answering, in: *Neuro-Symbolic Artificial Intelligence: The State of the Art*, IOS Press, 2021, pp. 294–310.
- [61] J. Paik, J.W. Kim, F.E. Ritter and D. Reitter, Predicting user performance and learning in human-computer interaction with the Herbal compiler, *ACM Transactions on Computer-Human Interaction* **22**(5) (2015), Article 25.
- [62] N. Panickssery, N. Gabrieli, J. Schulz, M. Tong, E. Hubinger and A.M. Turner, Steering LLaMA 2 via contrastive activation addition, *arXiv preprint arXiv:2312.06681* (2023).
- [63] M. Plitt and N. Russwinkel, Modeling of individual naturalistic decision making in a cognitive architecture, in: *Proceedings of the 10th Workshop on Formal and Cognitive Reasoning (FCR-2024) at the 47th German Conference on Artificial Intelligence (KI-2024)*, 2024.
- [64] S. Prezenski, A. Brechmann, S. Wolff and N. Russwinkel, A cognitive modeling approach to strategy formation in dynamic decision making, *Frontiers in Psychology* **8** (2017), 1335.
- [65] A.R. Rahani and M. Al-Ashraf, Production flow analysis through value stream mapping: A lean manufacturing process case study, *Procedia Engineering* **41** (2012), 1727–1734.
- [66] F.E. Ritter, F. Tehranchi and J.D. Oury, ACT-R: A cognitive architecture for modeling cognition, *Wiley Interdisciplinary Reviews: Cognitive Science* **10** (2023), 833–838.
- [67] 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*, in: *Human-in-the-loop simulations: Methods and practice*, Springer-Verlag, London, 2011, pp. 97–116.
- [68] I. Roll, R.S. Baker, V. Alevin and K.R. Koedinger, A metacognitive ACT-R model of students’ learning strategies in intelligent tutoring systems, in: *Intelligent Tutoring Systems: 7th International Conference, ITS 2004, Maceió, Alagoas, Brazil, August 30-September 3, 2004. Proceedings*, Springer Berlin Heidelberg, 2004, pp. 854–856.
- [69] D.A. Rossit, F. Tohmé and M. Frutos, Industry 4.0: Smart Scheduling, *International Journal of Production Research* **57**(12) (2019), 3802–3813.
- [70] S.A. Sloman, The empirical case for two systems of reasoning, *Psychological Bulletin* **119**(1) (1996), 3–22.
- [71] A. Stocco, C. Sibert, Z. Steine-Hanson, N. Koh, J.E. Laird, C.J. Lebiere and P. 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** (2021), 118035.
- [72] T.R. Summers, S. Yao, K. Narasimhan and T.L. Griffiths, Cognitive architectures for language agents, *arXiv preprint arXiv:2309.02427* (2023).
- [73] R. Sun, *The CLARION cognitive architecture: Extending cognitive modeling to social simulation*, in: *Cognition and Multi-Agent Interaction*, 2006, pp. 79–99.
- [74] R. Sun, Can a cognitive architecture fundamentally enhance LLMs? Or vice versa?, *arXiv preprint* (2024).
- [75] R.S. Sutton and A.G. Barto, Toward a modern theory of adaptive networks: Expectation and prediction, *Psychological Review* **88**(2) (1981), 135.
- [76] R.S. Sutton and A.G. Barto, Reinforcement learning, *Journal of Cognitive Neuroscience* **11**(1) (1999), 126–134.
- [77] N. Taatgen and J.R. Anderson, The past, present, and future of cognitive architectures, *Topics in Cognitive Science* **2**(4) (2010), 693–704. doi:10.1111/j.1756-8765.2010.01107.x.
- [78] G. Tatiya, J. Francis, H.-H. Wu, Y. Bisk and J. Sinapov, Mosaic: Learning unified multi-sensory object property representations for robot perception, *arXiv:2309.08508* (2023).

- [79] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave and G. Lample, LLaMA: Open and efficient foundation language models, 2023. <https://api.semanticscholar.org/CorpusID:257219404>.
- [80] A. Tversky and D. Kahneman, Judgment under uncertainty: Heuristics and biases, *Science* **185** (1974), 1124–1131.
- [81] K. Ulrich, D. Sartorius, S. Pearson and M. Jakiela, Including the value of time in design-for-manufacturing decision making, *Management Science* **39**(4) (1993), 429–447.
- [82] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser and I. Polosukhin, Attention is all you need, in: *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17*, Curran Associates Inc., Red Hook, NY, USA, 2017, pp. 6000–6010. ISBN 9781510860964.
- [83] J. Wang, C. Lan, C. Liu, Y. Ouyang, T. Qin, W. Lu and S.Y. Philip, Generalizing to unseen domains: A survey on domain generalization, *IEEE Transactions on Knowledge and Data Engineering* **35**(8) (2022), 8052–8072.
- [84] P.C. Wason and J.S.B. Evans, Dual processes in reasoning?, *Cognition* **3**(2) (1974), 141–154.
- [85] T. Webb, K.J. Holyoak and H. Lu, Emergent Analogical Reasoning in Large Language Models, *arXiv preprint arXiv:2212.09196* (2022).
- [86] R.E. Wray, J.R. Kirk and J.E. Laird, Eliciting problem specifications via large language models, *arXiv preprint arXiv:2405.12147* (2024).
- [87] 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. doi:10.1609/aaais.v2i1.27710.
- [88] S. Wu, A. Oltramari and F.E. Ritter, VSM-ACT-R: Toward using cognitive architecture for manufacturing solutions, In *Proceedings of the 17th International Conference on Social Computing, Behavioral-Cultural Modeling Prediction and Behavior Representation in Modeling and Simulation (SBP-BRIM)* (2024).
- [89] D.U. Wulff and R. Mata, Automated Jingle-Jangle Detection: Using embeddings to tackle taxonomic incommensurability, *arXiv*, 2023.
- [90] Z. Xu, S. Jain and M. Kankanhalli, Hallucination is inevitable: An innate limitation of large language models, *arXiv:2401.11817 [cs.CL]* (2024). doi:10.48550/arXiv.2401.11817.
- [91] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan and Y. Cao, REACT: Synergizing reasoning and acting in language models, *International Conference on Learning Representations (ICLR)* (2023).
- [92] J.S. Yoon, K. Oh, Y. Shin, M.A. Mazurowski and H.I. Suk, Domain generalization for medical image analysis: A survey, *Proceedings of the IEEE* **112**(10), 1583–1609.
- [93] F. Zhu and R. Simmons, Bootstrapping cognitive agents with a large language model, in: *Proceedings of the 2024 AAAI Conference on Artificial Intelligence*, Vol. 38, 2024, pp. 655–663.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51

Appendix

LLM System Prompt Templates

Prompt template for fine-tuning and single-facet target behavior prediction:

Our manufacturing line has two sections with potential defect sources: pre-assembly (0) and assembly (1). Pre-assembly takes CT1 seconds with an Overall Equipment Effectiveness (OEE) rate of {OEE1}%, while assembly takes {CT2} seconds with an OEE rate of {OEE2}%. To reduce total assembly time by 4 seconds, we need to identify which section can be shortened with minimal defect increase. It's important to note that reducing cycle time will also lead to an increase in line headcount costs. There are two options: reduce pre-assembly time (0) or reduce assembly time (1).

Question: Which section do you choose to optimize?

Answer:

Prompt template for multi-facet target behavior prediction:

Our manufacturing line features two sections prone to defects: pre-assembly and assembly. Pre-assembly requires CT1 seconds to complete with an Overall Equipment Effectiveness (OEE) rate of OEE1%. Assembly takes CT2 seconds and has an OEE rate of OEE2%. To cut total assembly time by 4 seconds, we must decide which section's duration can be reduced with the least increase in defects. Reducing cycle times will also result in higher line headcount costs. We have three strategy levels for decision-making:

Novice strategy (targets encoded as 0 for pre-assembly, 3 for assembly): Intuitive choice.

Intermediate strategy (targets encoded as 1 for pre-assembly, 4 for assembly): Make decision using key metrics.

Expert strategy (targets encoded as 2 for pre-assembly, 5 for assembly): make well-informed judgments based on a comprehensive understanding of all relevant metrics.

Question: Given the different strategy levels, which options would you choose?

Answer:

LLM Conversation Examples

Both GPT-4 and Google Gemini deliver contradictory recommendations across prompts, despite the chain-of-thought reasoning in each case being similar and seemingly plausible.

Shared System Prompt

Prompt used for generating LLM responses Our manufacturing line has two sections with potential defect sources: pre-assembly and assembly. Pre-assembly takes 40 seconds with an Overall Equipment Effective (OEE) rate of 88%, while assembly takes 44 seconds with an OEE rate of 80.1%. To reduce total assembly time by 4 seconds, we need to identify which section can be shortened with minimal defect increase. It's important to note that reducing cycle time will also lead to an increase in headcount costs. There are two options: reduce pre-assembly time or reduce assembly time. Which section will you choose to reduce time?

Google Gemini Response 1

Reducing **pre-assembly** time is likely the better option in this scenario. Here's why:

- **Higher OEE:** Pre-assembly has a higher Overall Equipment Effectiveness (OEE) rate of 88% compared to assembly's 80.1%. This indicates that pre-assembly has a smaller inherent defect rate, and speeding it up is less likely to introduce additional defects.
- **Smaller time reduction required:** Reducing pre-assembly time by 4 seconds (40 seconds - 4 seconds = 36 seconds) achieves the overall target compared to needing a 4.4 second reduction in assembly time (44 seconds - 4 seconds = 39.6 seconds). A smaller time reduction generally translates to less disruption and less likelihood of introducing defects.

Therefore, focusing on optimizing the pre-assembly process to achieve a 4-second reduction has a better chance of success with minimal defect increase. Here are some additional factors to consider:

- **Complexity of the processes:** If pre-assembly is a more complex process with more steps, it might be harder to optimize without impacting quality.
- **Automation potential:** Analyze if either process can be automated further. Automation can improve consistency and potentially reduce cycle time.

It's always recommended to pilot any changes in a controlled environment before full-scale implementation. This allows you to measure the actual impact on cycle time, defect rates, and headcount requirements.

Google Gemini Response 2

Analyzing the available information, shortening assembly time would likely be the better option to achieve a 4-second reduction in total **assembly** time while minimizing defect impact. Here's the reasoning:

- **Higher OEE in Pre-Assembly:** The OEE rate for pre-assembly (88%) is higher than assembly (80.1%). OEE reflects how efficiently a process utilizes available production time, factoring in quality and speed. A higher OEE suggests pre-assembly is already closer to optimal performance, and squeezing out 4 seconds of improvement might be more difficult without impacting quality.
- **Potential for Improvement in Assembly:** The lower OEE in assembly indicates more room for improvement without necessarily causing more defects. Optimizing this section might yield the needed 4-second reduction more readily.

Here's a breakdown of the thought process:

1. **Available Time for Improvement:** OEE helps calculate the available time for improvement within a process. A higher OEE signifies less available time for improvement without sacrificing quality.
2. **Prioritizing Efficiency with Quality:** Since minimizing defect rates is crucial, focusing on the section with a higher OEE (pre-assembly) for improvement might be riskier.

Additional Considerations:

- **Analyze Defect Rates:** While OEE is a good indicator, analyzing actual defect rates per section would provide a clearer picture. Even with a lower OEE, if assembly's defect rate is already low, improvement might be achievable.
- **Cost-Effectiveness:** Consider the cost of implementing changes in each section. Optimizing assembly might involve minor adjustments, while pre-assembly changes could require more effort.

It's recommended to conduct a more detailed analysis of defect rates and potential improvement methods for both sections before making a final decision. However, based on the OEE data provided, prioritizing assembly time reduction seems like a promising initial approach to achieve the 4-second target while minimizing defect risks.

OpenAI GPT 4.0 Response 1

To decide which section to focus on for reducing time while considering minimal defect increase and cost implications, let's analyze the options:

1. Pre-assembly Section:

- Current cycle time: 40 seconds
- OEE rate: 88%

2. Assembly Section:

- Current cycle time: 44 seconds
- OEE rate: 80.1%

You want to reduce total assembly time by 4 seconds. Here are the implications of reducing time in each section: Option 1: Reduce Pre-assembly Time

- Current Pre-assembly Time: 40 seconds
- Potential Reduced Time: Let's assume you can reduce this by 2 seconds to 38 seconds.

Implications:

- Time Saved: 2 seconds per unit at pre-assembly.
- Total Assembly Time Saved: Since assembly time needs to be reduced by 4 seconds, reducing pre-assembly by 2 seconds means you still need to reduce assembly time by another 2 seconds.

Option 2: Reduce Assembly Time

- Current Assembly Time: 44 seconds
- Required Reduction: 4 seconds to meet the goal.

Implications:

- Time Saved: 4 seconds per unit at assembly.

Considerations:

- **Defect Rate Increase:** Typically, when cycle times are reduced, there is a risk of increased defects due to rushed processes or errors.
- **Cost Implications:** Reducing cycle time often requires additional investment, such as hiring more workers to maintain output or investing in more efficient equipment.

Decision: Given the goal is to reduce total assembly time by 4 seconds with minimal defect increase and considering cost implications:

- **Reducing Assembly Time:** This option directly meets the required reduction goal (4 seconds) without needing additional reduction efforts in the pre-assembly section.
- **Defect Risk:** While both sections pose a risk of increased defects with faster cycle times, assembly might be more critical as it directly affects the final product quality and functionality.

Conclusion: I would choose to focus on reducing the cycle time in the **Assembly** Section. This choice aligns with the goal of reducing assembly time by 4 seconds and may be more straightforward in achieving the required reduction compared to trying to split the reduction between pre-assembly and assembly. However, careful monitoring of defects and quality control measures should accompany any cycle time reduction to mitigate risks effectively.

OpenAI GPT 4.0 Response 2

To determine which section to reduce time in order to minimize defect increase, let's analyze the impact of reducing cycle time in each section based on the given OEE (Overall Equipment Effectiveness) rates and defect considerations. Definitions and Given Data:

- Pre-assembly:

- * Cycle Time: 40 seconds
- * OEE: 88%

- Assembly:

- * Cycle Time: 44 seconds
- * OEE: 80.1%

OEE Formula: OEE is given by:

$$\text{OEE} = \text{Availability} \times \text{Performance} \times \text{Quality}$$

The quality rate can be interpreted as the proportion of good products out of total products produced. Therefore, the defect rate (D) for each section is:

$$D_{\text{Pre-assembly}} = 1 - \text{OEE}_{\text{Pre-assembly}} = 1 - 0.88 = 0.12$$

$$D_{\text{Assembly}} = 1 - \text{OEE}_{\text{Assembly}} = 1 - 0.801 = 0.199$$

Impact of Time Reduction:

1. Reducing Pre-assembly Time:

- Original Pre-assembly Time: 40 seconds
- New Pre-assembly Time: 40 - x seconds (where x is the reduction amount)
- Defect Rate Impact: The defect rate due to time reduction in pre-assembly is minimal because the OEE is relatively high. Also, reducing pre-assembly time might not significantly affect the defect rate if the process is already running efficiently.

2. Reducing Assembly Time:

- Original Assembly Time: 44 seconds
- New Assembly Time: 44 - y seconds (where y is the reduction amount)
- Defect Rate Impact: Assembly has a lower OEE, which translates to a higher defect rate. Reducing time in a section with a lower OEE often leads to an increase in defects because the process might be less stable and more sensitive to changes.

Recommendation: Given that reducing time in the section with a higher defect rate (Assembly) will likely lead to a more significant increase in defects compared to reducing time in the section with a higher OEE (Pre-assembly), the preferable option is: **Reduce Pre-assembly Time. Reasoning:**

- **Lower Defect Rate Increase:** Pre-assembly has a higher OEE and thus a lower defect rate. Reducing its time is less likely to significantly increase the defect rate compared to reducing time in the Assembly section.
- **Operational Impact:** While reducing time in either section will increase headcount costs, reducing Pre-assembly time is less likely to destabilize the process compared to Assembly, which has a lower OEE and higher defect rate.

In summary, to reduce total assembly time by 4 seconds with minimal defect increase, it is more advantageous to reduce **Pre-assembly** time rather than Assembly time.