UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

School of Informatics

Bayesian Inference of Capability Profiles for Machine Common Sense

End of Degree Project

Bachelor's Degree in Data Science

AUTHOR: Sanchez Garcia, Pablo

Tutor: Hernández Orallo, José

ACADEMIC YEAR: 2023/2024

Resumen

Los sistemas de IA son generalmente evaluados para entender su rendimiento empleando una variedad de 'benchmarks', sirviéndose de una única métrica para describirlo, lo cual nos provee una imagen muy simple de sus capacidades. Sin embargo, esta metodología no es adecuada cuando queremos entender su rendimiento en entornos de una naturaleza más genérica.En este proyecto, tomaremos datos de rendimiento de algunos de los agentes desarrollados en el proyecto MCS de DARPA e inferiremos sus perfiles de capacidad mediante la triangulación bayesiana proporcionada por la metodología Measurement Layouts. Éstos, al fin y al cabo son redes bayesianas semánticamente ricas inferidas mediante el motor probabilístico ofrecido en la librería PyMC, la cual se encuentra disponible en Python. En el proyecto buscamos extraer perfiles de capacidad mediante datos exhaustivos provenientes de agentes diseñados con el objetivo de mostrar capacidades de 'sentido común'. Todo ello mediante inferencia bayesiana. Igualmente analizaremos el poder predictivo y explicativo de esta técnica, comparándola así con métodos más tradicionales como la simple obtención de métricas de precisión a partir de 'benchmarks' masivos, o la simple predicción basada en métricas agregadas de rendimiento.

Palabras clave: Evaluación IA, Redes Bayesianas, Inferencia Bayesiana, PyMC, Evaluación de Capacidades, Benchmarks, Perfil de Capacidad, Sentido Común

Resum

Els sistemes de IA són generalment avaluats per a entendre el seu rendiment emprant una varietat de 'benchmarks', servint-se d'una única mètrica per a descriure'l, la qual cosa ens proveïx una imatge molt simple de les seues capacitats. No obstant això, esta metodologia no és adequada quan volem entendre el seu rendiment en entorns d'una naturalesa més genèrica. En aquest projecte, prendrem dades de rendiment d'alguns dels agents desenvolupats en el projecte MCS de DARPA i inferirem els seus perfils de capacitat mitjançant la triangulació bayesiana proporcionada per la metodologia Measurement Layouts. Estos, al cap i a la fi són xarxes bayesianes semànticament riques inferides mitjançant el motor probabilístic oferit en la llibreria PyMC, la qual es troba disponible en Python. En el projecte busquem extraure perfils de capacitat mitjançant dades exhaustives provinents d'agents dissenyats amb l'objectiu de mostrar capacitats de 'sentit comú'. Tot això mitjançant inferència bayesiana. Igualment analitzarem el poder predictiu i explicatiu d'esta tècnica, comparant-la així amb mètodes més tradicionals com la simple obtenció de mètriques de precisió a partir de 'benchmarks' massius, o la simple predicció basada en mètriques agregades de rendiment.

Paraules clau: Avaluació IA, Xarxes Bayesianes, Inferència Bayesiana, PyMC, Avaluació de Capacitats, Benchmarks, Perfil de Capacitat, Sentit Comú

Abstract

AI systems are usually evaluated with a variety of benchmarks to determine their performance for specific tasks, using a single metric which provides a simplistic image of their capabilities. However, this procedure is insufficient when we want to evaluate and infer their capabilities in more general settings. In this project, we will take performance data from some of the agents that were developed in the DARPA's MCS project and infer their capability profiles through Bayesian triangulation provided by the Measurement Layouts methodology. These are, semantically-rich hierarchical Bayesian networks (HBN) that are inferred using the probabilistic programming engine PyMC, which is available in Python. Using extensive data of several agents that were tasked to solve a variety of common-sense problems, we can extract their capability profiles and compare them with each other just by using Bayesian triangulation. We analyse the predictive and explanatory power of the inferred Bayesian models to evaluate AI over other procedures like just estimating the aggregate accuracy of the agents with massive benchmarks.

Key words: AI Evaluation, Bayesian Networks, Bayesian Inference, PyMC, Capability-oriented Evaluation, Benchmarks, Capability Profile, Common Sense

Contents

List of Figures

List of Tables

List of algorithms

CHAPTER 1
Introduction

Computers, and by extension AI systems, are often defined as universal machines [67]. Recent events have demonstrated "their potential extends to tackle a boundless universe of tasks" [65]. The introduction of Transformers [69] back in 2017 and the subsequent development of large-scale pre-trained versions of them [51, 52, 10]; inaugurated the era of general-purpose AI systems, also known as Foundation Models. Here, the term *general* addresses not only the demonstration of excellence in handling multimodal information like audio, video, image or text [76, 64]; but also to the adaptability through in-context learning (ICL) abilities [22], which enable them to thrive in unseen scenarios during their training.

While the notion of AI systems reaching superintelligence –AI being superior to human in "practically every field"– in the short-term is still a controversial discussion in this community [7], there is an overall consensus on the potential variety of intrinsic and extrinsic catastrophic risks posed by highly-capable foundation models [6, 30].

In response, many experts argue that due to the distinct display of intelligence from these systems, compared to natural (animal) intellect, we should reference cognitive science and adopt universal psychometrics –"the analysis and development of measurement tools for the evaluation of behavioural features in the machine kingdom, including cognitive abilities and personality traits"– [32] as the preferred schema for evaluating these general-purpose systems [71].

This represents a paradigm shift from the current evaluation philosophy, sometimes referenced to as "task-oriented evaluation", which primarily tests systems on *gigantic* diverse benchmark suites [61] to compute an aggregate performance score. While this metric compresses a general –but superficial– view of systems' performance, allowing an straightforward comparison between systems, it is only a measure of how a system performs according to a distribution of items. "When this distribution changes, performance also does" [11]. Therefore, it is not a suitable approach for evaluating general-purpose agents.

This project presents an application of a solution to current AI evaluation intricacies: the *Measurement Layouts* [12]. This cognitive approach to AI evaluation relies on extensive experimental data from agents and (Bayesian) triangulation [29] to infer the cognitive profile of general-purpose systems. This allows not only understanding what a system is capable of, but also to predict future performance and comprehend the nuances of its behaviour by mapping test items with their cognitive demands to capabilities (abilities or skills). This project has built upon the RECOG-AI[1] initiative, a multidisciplinary initiative from the Leverhulme Centre for the Future of Intelligence[2] proposed to "provide

[1] http://lcfi.ac.uk/projects/kinds-of-intelligence/recog-ai/
[2] http://lcfi.ac.uk/

a framework and benchmarks for measuring the capabilities of AI systems". Specifically, we will apply this methodology to common sense agents from DARPA's Machine Common Sense program (MCS)[3]. More specifically, we will take performance data from some of the agents that were developed in this project and infer their capability profiles through Bayesian triangulation provided by the Measurement Layouts methodology.

1.1 Objectives

This project pursues the following main objectives:

1. Infer capability profiles of DARPA's Machine Common Sense agents using Measurement Layouts and provide comprehensive comparison of their common sense capabilities.
2. Assess the predictive performance of Measurement Layouts in estimating the performance of agents on unseen data, and compare it to other methods such as making use of assessors or naive predictions such as using observed aggregate performance as the indicator for future performance.

Secondary objectives include:

1. Understand how the Hierarchical Bayesian Networks (HBN) can be applied to model cognitive processes.
2. Review current AI evaluation methods and identify their weaknesses.
3. Highlight the need for redefining evaluation techniques for general-purpose learning agents.
4. Explore the potential of cognitive modelling and cognitive compared computation in AI Evaluation.
5. Integrate advanced Monte Carlo simulation algorithms for training Bayesian hierarchical models.

1.2 Memory Structure

The present memory has the following structure:

In the Related Work and Background Chapter, we provide an overview of current AI systems, the risks of highly-capable AI, and the importance of evaluation to mitigate these risks. We discuss current AI evaluation dynamics and propose a "redirection". We introduce the concept of common sense and its relation to DARPA's Machine Common Sense (MCS) Programme, and explain how Bayesian Theory can model cognitive mechanisms similar to our AI evaluation methodology.

In the Materials and Methods Chapter, we delve into DARPA's MCS Programme, explaining the simulation tools that were used and the performance data annotation process by the MCS Evaluation team. We then present the statistical and probabilistic framework for the measurement layouts, including Bayesian Networks, Hierarchical Bayesian Networks, and Markov Chain Monte Carlo algorithms. The chapter concludes with an introduction to the measurement layouts and their Bayesian formulation.

[3] https://www.darpa.mil/program/machine-common-sense

In the Exploratory Analysis Chapter, we analyze MCS Evaluation Data to identify properties predictive of agents' performance and build intuition about their capabilities.

In the Experimental Setting Chapter, we describe the experimentation phase, metrics for evaluating measurement layouts' predictive performance, and comparison approaches. We also present the general topology of the measurement layouts used.

In the Results Chapter, we introduce the two specific settings of the measurement layouts and compare the inferred capability profiles and predictive power across different settings, granularities of evaluation data –instance and aggregated level–, and agents.

The project concludes by discussing limitations, future steps, the degree of achievement of objectives, and the influence of the Bachelor's Degree Programme competences on the project.

1.3 Collaborations

This project represents the continuation from the work put by the RECOG-AI Team at the Leverhulme Center for the Future of Intelligence, at the University of Cambridge onto the MCS Programme from DARPA. They pursued the "Robust Evaluation of Cognitive Capabilities and Generality in Artificial Intelligence". The exchange of opinions with them has been crucial to shape this initiative, providing an introduction to the measurement layouts methodology, assistance with its formalisation and the intuitions to properly make use of the framework for the purpose of this project.

On the other hand, it was worked together with the MCS Programme Evaluation team, composed by Professor David Moore, Professor Koleen McCrink and Professor Lisa Oakes. They provided a comprehensive explanation of how the evaluation process was carried out, including the annotation phase of evaluation data, a crucial step for our endeavour.

Both teams contributed to this project by proportioning myself with pivotal theoretical foundations on cognitive science and development psychology, domains that are the core of this project.

CHAPTER 2
Related Work and Background

2.1 Categorisation of AI systems

The present project embraces the discipline of universal psychometrics, defined as the "measurement of cognitive abilities for the *machine kingdom*" [32, 33]. The latter concept addresses the "set of all interactive systems taking inputs and producing outputs, possibly asynchronously, through interfaces, bodies, sensors and actuators, etc.". This ranges from all types of biological life (human and non-human animals) to artificial life.

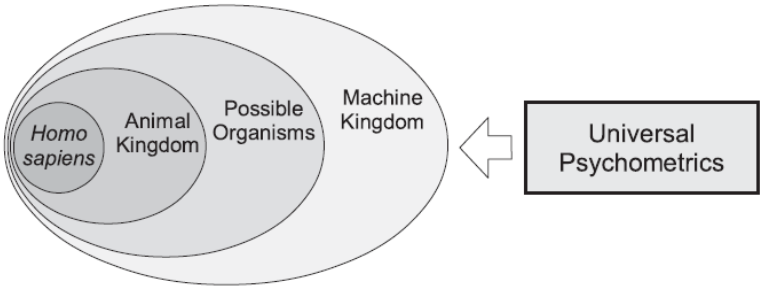

Figure 2.1: Euler Diagram representing the groups that integrate the Machine Kingdom. Figure taken from [32]

The reason for the existence of universal psychometrics specially arises from the integration of artificial life into the population of interest for evaluating and analysing intelligent behavior. We can draw the proposal from [32] to understand this population, as offers a simple yet comprehensive categorisation of the systems that are included of this group:

1. Computers: group composed of "any type of computational behaviour, including any artefact that is designed with some kind of artificial intelligence".

2. Cognitively enhanced organisms: by cognitive enhancing, it is referred to how an organism (human or not human) can "get around" [50] or alter its cognitive abilities through "cognitive extenders". These are "external physical or virtual elements that are coupled to enable, aid, enhance, or improve cognition, such that its effect is lost when the element is not present".[34] An example of this is the case when a human has access to GPS or translation tools.

3. Biologically enhanced computers: when computers need of humans to achieve certain tasks.

4. Hybrid collectives: groups of organisms from the machine kingdom, not necessarily belonging to the same "species".

This project puts the focus on the former group, specially in artificial intelligence agents. An agent could be considered anything –a program, a robot– that reacts to some stimuli –basically, data– provided by the environment it resides in. Taking the taxonomy provided by S. Russell and P. Norvig in [54], we have the following types of agents:

1. Simple agents: they ignore previous received stimuli/data to react to the current input.
    - Example: Basic AI in video games that reacts to the player's actions without considering the history of interactions.
2. Model-based agents: they have some knowledge of the world/environment they reside in. They update the 'model' they have of the world as they perceive changes.
    - Example: Thermostats that adjust heating based on current temperature and historical data to maintain a comfortable environment.
3. Goal-based agents: they have a description of the final states/situations that are desirable and, as a consequence, should be pursued. When "satisfactory results can come from a single action", we call it a *search* problem; while when it requires long sequences of decisions it is a *planning* problem.
    - Example: In the entertainment industry, goal-based systems agents can be used to suggest content that resonates with a target audience.
4. Utility-based agents: similarly to goal-based agents, their decisions are made on a basis of the improvement of a performance measure, that determined how desirable the current or next state is. To measure this, agents incorporate a *utility function* which is an "internalisation of the performance" measure. These agents try to optimised depending on the expected utility of their actions outcomes.
    - Example: Trading algorithms in financial markets that decide to buy or sell stocks based on the expected utility (profit) of the transactions.
5. Learning agents: These agents can learn from their their actions and improve their performance over time by adjusting their behaviour based on past experiences. This is the type of agents in current state-of-the-art systems.
    - An example of this agents are Non-Playable-Characters (NPC) in some video games, which are able to learn from their previous "plays" and strategies

As mentioned above, a great part of the most advanced technologies in the field are applications derived from machine learning. Usually, taxonomies for categorising AI learning agents classify them based on the training technique employed –e.g. supervised learning, reinforcement learning–; the specific problem their intended for –e.g. object segmentation, sentiment analysis–; the nature of the task –e.g. classification, regression, generation, etc.–; but in this case, we will provide some examples –without loss of generality– of AI systems that are a matter of interest for the AI Evaluation discipline.

1. Reinforcement Learning Agents: they are a distinctive class of learning agents that learn by interacting with their environment and receiving feedback in the form of

rewards or penalties. This "feedback loop" allows the agent to learn optimal behaviours to achieve its goals. The core components of reinforcement learning are: the environment; the state of the environment; the actions, which are the set of possible moves of the agent; the reward, that is the feedback provided to the agent after taking an action or a sequence of them; a policy, which is the strategy that the agent is pursuing to guide its actions given the feedback/reward received; and the value function, that estimates the expected reward from a given action in a specific state.

This specific agents training schema derives from the concept of "conditioning". This terms comes from psychology and explains how animals have a predisposition against stimuli, and how the actions derived from these stimuli can be conditioned –with rewards or penalties– to favour a desired behaviour [53, 32].

The field of reinforcement learning has benefited from the integration of deep neural networks in these systems. This had a considerable impact on the field as "representation learning with deep learning enables automatic feature engineering and end-to-end learning, so that reliance on domain knowledge is significantly reduced or even removed".[43]

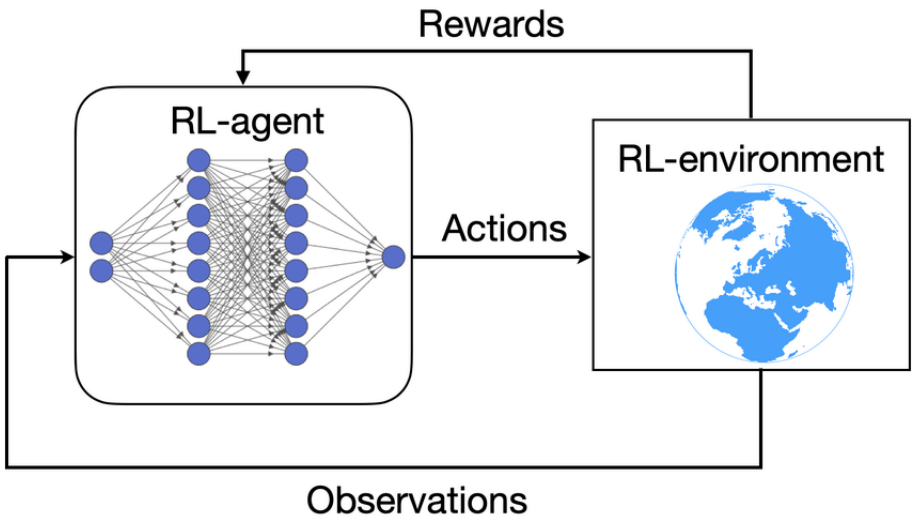

Figure 2.2: The workflow of deep reinforcement learning. Figure taken from [37]

Deep reinforcement learning (DRL) has contributed to the creation of very complex cognitive systems, ranging in the field of application from natural language processing to robotics. These systems exhibit very distinct intelligent behaviour than humans or animals, and have accelerated the arrival of general-purpose agents. The characterisation of these systems capabilities is one the biggest challenges in the field nowadays, given their transformative yet risky potential[6]. Reinforcement learning represents a paradigm for the understanding of cognitive processes for every individual that can be simulated in a general setting where they can interact with their environment through the use of observations, actions and rewards.

Some illustrative examples of deep reinforcement learning agents are *Tesla's* self-driving cars; DeepMind's AlphaGo [59], that learnt to play go and beat professional players; or DeepMind's Agent57, which outperforms humans in 57 Atari games [3].

2. Transformers: these are one the most promising deep learning models, and have been widely introduced in many contexts. They were introduced in 2017 [69], and they leverage attention mechanisms in a similar way cognitive systems do, allowing the selective processing and learning from what is considered "relevant information" [49, 9]. They are adopted because of their efficiency and versatility, not only

for the range of fields and problems they can be used in, but also for the variability of usage they have. They can be used as decoders, encoders or both at the same time.

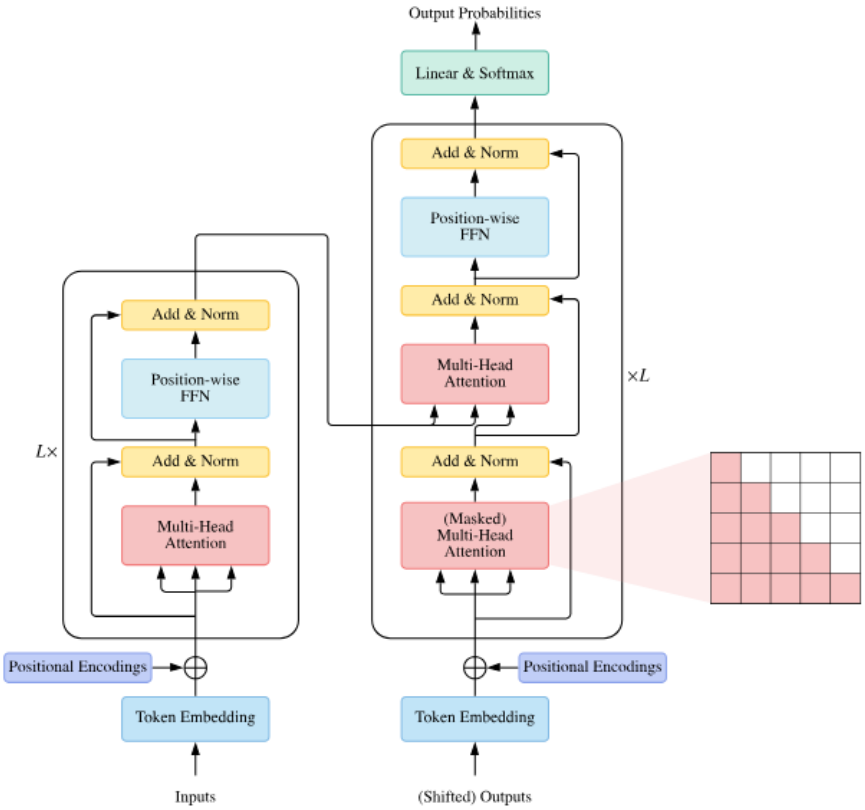

Figure 2.3: Architecture of *Vanilla Transformer*, taken from [49]

[49] proposes a taxonomy for transformers attending to the "variants at the module level, the architecture level, the pre-training schema and the applications". However, we will focus on generative pre-trained transformers, which are the building blocks for current Large Language Models (LLMs), the first example of general-purpose agents of our time.

Generative pre-trained transformers (GPT) [51, 52, 10] are transformers trained on large amounts of text following the language modelling approach, taken from Language Models (LM), a probabilistic approach of modelling a language which basically consists on "given a sequence of words, infer which is the most likely word to come after". GPTs are also named Large Language Models.

The remarkable potential of these systems arises from the capabilities that emerge from such a *simple* learning process [72], allowing them to achieve remarkable performance across a multitude of diverse downstream tasks and applications, and excelling at few-shot learning settings –"constructing new knowledge from sequences of labelled examples presented in the input without further parameter updates" [1].

Generative pre-trained transformers are also the pillars for building Vision-Language Models, an extension of Large Language Models in which architectural modifications are introduced to handle multimodal information [76, 64].

Large Language Models –and therefore VLLMs too–, also called "Foundation Models", represent the latest evolutionary stage of a process of "emergence and homogenisation"

[5] of machine learning agents over the last decades. These systems are the first example of general-purpose agents in the history of Artificial Intelligence, and, despite the vast number of opportunities they pose, their emergent nature and the very distinct display of intelligent behaviour results in an immanent challenge of understanding and evaluating their capabilities. [21, 5] Their ability to perform well across various tasks without task-specific training demonstrates a form of general intelligence, a key focus of universal psychometrics.

2.2 AI Safety

Sooner or later, artificial superintelligence will arrive. Nick Bostrom defines these systems in the following way: "AI being superior to human in practically every field" [6].

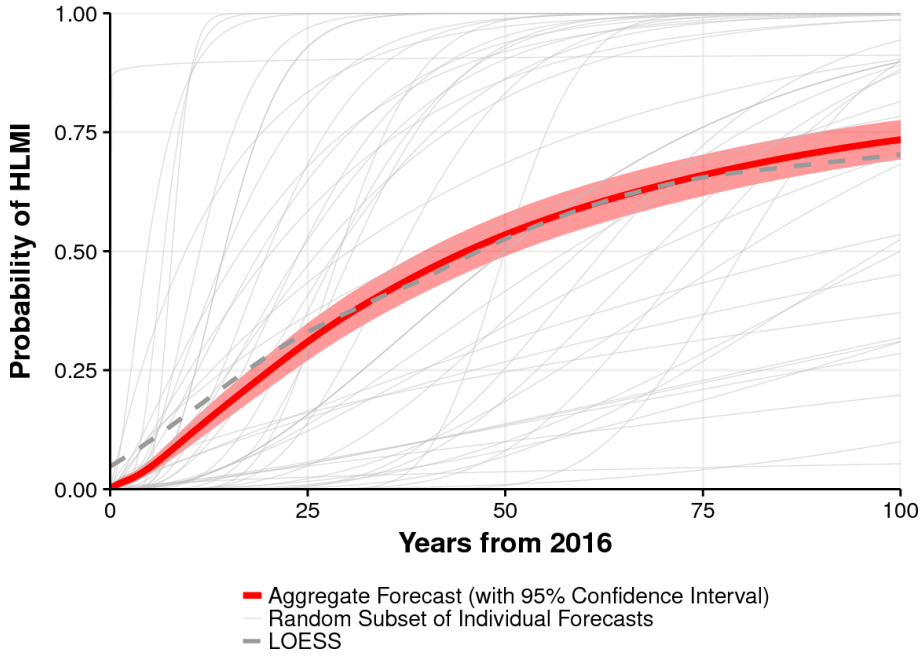

Figure 2.4: Aggregate subjective probability of 'AGI-level machine intelligence' arrival by future years. Prediction comes from 352 researchers who published at the 2015 NIPS and ICML. Figure taken from [26]

The arrival of a system with this level of intelligence may trigger the hypothetical "technological singularity" event –humans losing control of AI [75]– and many of the catastrophic risks associated to it [30].

The Center for AI Safety (CAIS)[1], one of the leading AI research laboratories provides a comprehensive view of the catastrophic risks from artificial superintelligence in [30]. They categorise these risks into four groups, leading to think that risks can be materialised as "an intentional cause, environmental/structural cause, accidental cause, or an internal cause". In any case, their work presents an incentive for developing better evaluation tools that allow us to know which are the actual capabilities of these systems. In this sense, well-reputed AI safety "thinker" and researcher Roman Yampolskiy states in [74] that potential risks and catastrophes associated with AI systems have historically proved to be *proportional* to the causing systems' capabilities.

Having said this, AI Safety can be defined as the area of research which is focused on decreasing the expected/possible risks from AI systems. This definition might seem very

[1] https://www.safe.ai/

vague due to that AI Safety is a very heavily loaded concept and it covers a wide range of subfields. We can have a better idea of what actually AI Safety intends to by looking at some of the areas it brings together:

1. Alignment: Paraphrasing the definition given by Paul Christiano, “when it is said that say system A is aligned with an operator H, it is meant that A is trying to do what H wants it to do” [18]. Many experts depict a future where we will delegate so many responsibilities to highly-capable AI systems. If these systems are not doing what we intend them to do, the results could be catastrophic. We can break down the alignment field into two problems:

2. Moral philosophy is one of the most complex areas of AI Safety, it is basically a debate about morality on giving arguments for the definition of “good” policies that in the case of being learnt would lead to “good” outcomes. This discussion is widely covered and explained in [8].

3. Competence, which targets AI to effectively accomplish the tasks it is designed for. This field might seem trivial, but there are many recent examples that highlight the relevance of it. [2] [3] [4]

4. Governance. Given the high-stakes implications of advanced AI in every imaginable scope/field, governance seeks for investigating “how humanity can best navigate the transition to advanced AI systems” [20] and regulating when the development or deployment is potentially harmful at a societal scale.

While the accelerated process of building general-purpose AI is motivated by the beneficial potential of this technology, as we have pointed out previously, this may lead general-purpose agents to acquire capabilities that pose “extreme risks”. Recent work promoted by DeepMind gathered many experts view on the pivotal role of model evaluation for the “identification of dangerous capabilities and the propensity of models to apply their capabilities for harm” [58]. One of the key arguments that was discussed was the unpredictable nature of foundation models emergent abilities –abilities which are not present in small models but are present in larger ones–, that Large Language Models are depicting [72]. These also include harmful capabilities that their developers did not aim for [25]. They state that a model should be considered as highly dangerous "if its capability profile is sufficient for extreme harm in the case of misuse or misalignment". This is why the project that is being presented with this thesis has such a possible beneficial impact, as it allows inferring robustly AI systems capability profiles [12].

Also, a recent investigation led by multiple Turing Award winners proposes an alternative approach to AI safety development referred to “guaranteed safe” AI [21]. It consists on "providing the sufficient mechanisms in AI design and deployment to ensure high-assurance quantitative safety guarantees". They point out that these mechanisms/measures can be summarised into three crucial components: (a), "a world model that provides explanation of how AI is affected by the outside world"; (b) "a safety specification to state mathematically which effects/behaviour are acceptable"; and (c), "a verifier to provide proof of AI satisfying the safety specification relative to the world model".

Despite this philosophy differing from the approach this project presents for AI evaluation, there is a consensus on the urgency of redesigning AI evaluation tools. Similar

[2] https://www.wired.com/story/zillow-ibuyer-real-estate/

[3] https://www.euractiv.com/section/disinformation/news/youtubes-algorithm-fuelling-harmful-content-study-says/

[4] https://www.theverge.com/2018/7/26/17619382/ibms-watson-cancer-ai-healthcare-science

to the conclusions drawn from [5], it is agreed that current evaluation procedures cannot provide a comprehensive view of advanced AI capabilities. The latter proposes that understanding foundation models' behaviour may require of a multidisciplinary approach so that "evaluation tools are precise in terms of *what* is actually being assessed" with each of items the environment/benchmark/test data may be composed by. This is the spirit of the Measurement Layouts [12].

2.3 AI Evaluation: The Paradigm Shift

Now that we are aware of the relevance of evaluating AI systems due to not only their potential risks, but also to the fact that they are increasingly being introduced in high-stakes situations, let's discuss the "ongoing" practices for evaluating them.

Current evaluation standards mainly rely on using experimental benchmarks, summarising with a metric –e.g. error rate, accuracy, correlation coefficient, Brier score, etc.– "how well" a given system performs. This tendency has induced a dynamic in which advancements in any sub-field of AI to be considered as promising must be accompanied of a "SOTA" jump in any benchmark –i.e. a substantial improvement in the performance metric of the benchmark [44]. This fosters that research can be sometimes focused on improving model's performance on benchmarks, and that systems' capabilities are regarded and compared on the "linear order" the evaluation metric provides [57], what eventually leads research to be focused on "overfitting" to the benchmark dataset. Illustrative of this phenomenon are the leaderboards' plots provided by Papers with Code [5]. In Figure 2.5 we see results for Natural Language Inference from the GLUE benchmark, "a collection of tools for evaluating the performance of models across a diverse set of existing NLU tasks" [70]

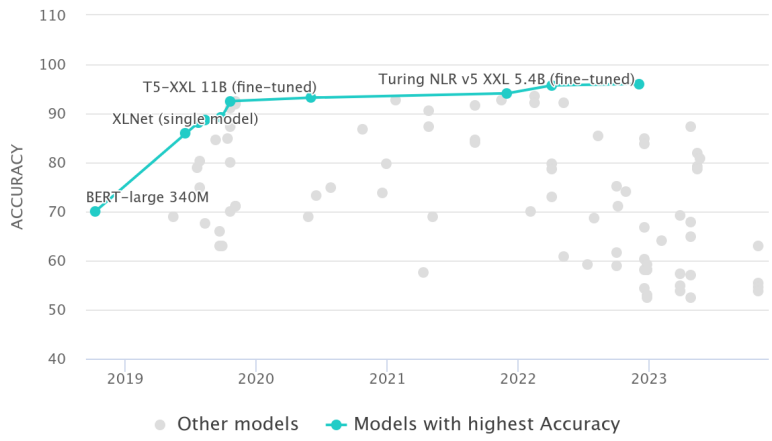

Figure 2.5: Natural Language Inference Task Performance from GLUE Benchmark

Then, when performance is saturated, benchmark creators launch a new set of samples updating the dataset, claiming that it is more challenging. However, there is some misconceptions about this cycle; improvement on benchmark results do not imply systems are not necessarily becoming more capable, those more challenging samples are basically new, and given that benchmarks only measure competence in a distribution of items, it is very likely that more variability may bring out more failures. Nevertheless, models reaching superhuman level in a given benchmark does not mean that it is more

[5] https://paperswithcode.com/

capable than human, instead, it outperforms human in that specific distribution of items. This dynamic is sometimes called "challenge-solve-and-replace" [57, 31].

Moreover, despite aggregate metrics providing an overview of systems' performance, this task-oriented approach usually omits details on which kind of instances systems struggle or succeed, information that taking a more curated procedure, trying to annotate demands from instances could provide a very valuable insight into systems' capabilities. Even when there is an intention to create a variate benchmark composed by many tasks that can potentially measure different capabilities, like in the case of Beyond the Imitation Game (BIG-Bench) [61], developers opt for reporting results with aggregate metrics. Another recent example of this takes place when AI Research companies report the results of their latest Large Language Model. They take a "bunch" of benchmark datasets, they check whether they are better than competence and they publish the results they are interested in. An example of this can be seen in Figure 2.6.

	Claude 3 Opus	Claude 3 Sonnet	Claude 3 Haiku	GPT-4	GPT-3.5	Gemini 1.0 Ultra	Gemini 1.0 Pro
Undergraduate level knowledge *MMLU*	**86.8%** 5 shot	79.0% 5-shot	75.2% 5-shot	86.4% 5-shot	70.0% 5-shot	83.7% 5-shot	71.8% 5-shot
Graduate level reasoning *GPQA, Diamond*	**50.4%** 0-shot CoT	40.4% 0-shot CoT	33.3% 0-shot CoT	35.7% 0-shot CoT	28.1% 0-shot CoT	–	–
Grade school math *GSM8K*	**95.0%** 0-shot CoT	92.3% 0-shot CoT	88.9% 0-shot CoT	92.0% 5-shot CoT	57.1% 5-shot	94.4% Maj1@32	86.5% Maj1@32
Math problem-solving *MATH*	**60.1%** 0-shot CoT	43.1% 0-shot CoT	38.9% 0-shot CoT	52.9% 4-shot	34.1% 4-shot	53.2% 4-shot	32.6% 4-shot
Multilingual math *MGSM*	**90.7%** 0-shot	83.5% 0-shot	75.1% 0-shot	74.5% 8-shot	–	79.0% 8-shot	63.5% 8-shot
Code *HumanEval*	**84.9%** 0-shot	73.0% 0-shot	75.9% 0-shot	67.0% 0-shot	48.1% 0-shot	74.4% 0-shot	67.7% 0-shot
Reasoning over text *DROP, F1 score*	**83.1** 3-shot	78.9 3-shot	78.4 3-shot	80.9 3-shot	64.1 3-shot	82.4 Variable shots	74.1 Variable shots
Mixed evaluations *BIG-Bench-Hard*	**86.8%** 3-shot CoT	82.9% 3-shot CoT	73.7% 3-shot CoT	83.1% 3-shot CoT	66.6% 3-shot CoT	83.6% 3-shot CoT	75.0% 3-shot CoT
Knowledge Q&A *ARC-Challenge*	**96.4%** 25-shot	93.2% 25-shot	89.2% 25-shot	96.3% 25-shot	85.2% 25-shot	–	–
Common Knowledge *HellaSwag*	**95.4%** 10-shot	89.0% 10-shot	85.9% 10-shot	95.3% 10-shot	85.5% 10-shot	87.8% 10-shot	84.7% 10-shot

Figure 2.6: Report from Claude 3: A Large Language Model from Anthropic. Figure taken from Anthropic's Claude 3 report

Once again, the reason why task-oriented evaluation based on reporting aggregate results from massive benchmark does not seem appropriate relies on the fact that "aggregate metrics depend not only on the capability of the system but also on the characteristics of the instances used for evaluation" [13]. At the end, this metric represents the success degree in a particular distribution of items. Moreover, while it may be claimed that a "sufficiently variate" set of datasets in terms of the tasks/domains could provide a meaningful insight into a system's "general capabilities", recent work has proved a model can excel on two considerably distinct benchmarks by leveraging the same underlying capabilities [46].

When it comes to characterising general-purpose agents –defining them as a system that can do a range of tasks for which it has not been trained/prepared–, it seems that a feature/capability-oriented approach could make the difference. While task-oriented approach can measure to which extent a system excels at particular task, capability-oriented

evaluation intends to infer some agent's features –capabilities and personality traits. This could allow us to build constructs –a set of factors of features that explain great part of an individual's behavioural "variance" [55]– that are predictive and explanatory about systems behaviour. This approach is inspired by psychometrics, a discipline this project embraces.

2.4 Cognitive Evaluation of AI and the Challenge of Measuring Capabilities

As it has been pointed out above, evaluating systems in terms of capabilities is not only a more comprehensive approach for understanding AI systems intelligent behaviour, but "the way" to provide high-quality assurance of advanced AI agents safety or alignment.

Nonetheless, this poses some challenges, and the best way to understand them is by going through some of the ideas discussed in [2]. In this work it is reviewed some of the complexities of evaluating Foundation Models capabilities. Given the "general" nature of this systems, we can extrapolate some conclusions that were drawn for the capability-oriented evaluation field. Some of the key points are:

1. Despite having similar performance in some tasks, the capabilities from AI and humans are very likely to be "mechanistically and behaviourally distinct". This phenomenon is also referred to as humans and AI having "different capability shapes".
2. Given this immanent and undoubted difference in capabilities, we should be cautious extrapolating human intelligence evaluation procedures and concepts to AI Evaluation. Therefore, constructs inferred about natural (human and non-human) intelligence to understand organisms capabilities "may be ill-suited" to describe AI capabilities.
3. The term "capability" has been used indiscriminately to address "models being able to perform well on tasks of some particular type". It suggests some ideas to (re-)conceptualise capabilities.
4. It puts as examples of conceptualisation, some psychometrics' techniques that can be applied to systems to infer "factors" –in the case of factor analysis [15]– or latent variables that can explain "measurements across subjects". In this case, capabilities would be those inferred factors.
5. One of the most relevant properties of a conceptualisation of capabilities is that it must ensure that we can make robust claims about the presence of capabilities –capabilities are present to some extent, or absent–.

Moreover, the main problem that has originated undesired dynamics on task-oriented evaluation [57] we discussed in Section 2.3 comes from "using human intelligence as a yardstick, what limits our vision of what AI should be, how to devise benchmarks and how to extrapolate beyond them" [31]. Given the potential of AI reaching human level in many domains, if we want to robustly evaluate these systems, there is a need to change the yardstick for devising evaluation procedures [31]. Proposes to break down evaluation domains into dimensions that allow evaluators to introduce "cognitive modifications" so that the "the space of evaluation stretches longer and wider than the trajectory that is defined by humans". Also, it discusses how relevant it is to map that space of dimensions, to a scale with its respective units for measuring them, claiming that the difficulty of the item in the given dimension should be a fundamental principle for the definition of the

units. In this sense, systems capabilities would be defined by "the level of difficulty that can be achieved by a system".

Following this idea, [35] devises a metric to evaluate *generality*, considered to be "comprehensive performance up to a certain level of difficulty and capability". It is claimed that the degree of generality depends on how capability is displayed as a function of the task difficulty. This proposal aims to provide an alternative to evaluate general intelligence without the need to rely on populational variance, as it has been usually done in the field of psychometrics with the *g factor* Spearman "discovered" that could explain most intelligence test results [60].

Another approaches inspired in psychometrics that have been used to evaluate AI capabilities are Factor Analysis [40, 14]; Item Response Theory [45]; or Structural Equation Modelling [68]. The latter has the inconvenience of relying on unlikely premises like linear relationships and normally distributed means of capabilities. In the case of Factor Analysis and Item Response Theory, they are also populational, and derive abilities and difficulties relative to population averages, therefore the parameters of the same item can change if we add new items, or the abilities of a system may change if we add more systems, and this is an issue for AI, where systems are rarely stable in number and behaviour.

In [71], there is a call for adapting universal psychometrics to evaluate general-purpose because unlike task-oriented evaluation, psychometrics focuses on latent constructs which provide predictive and explanatory power, essential for understanding AI agents' behaviour and even for improving their performance. It is stated that a rigorous evaluation procedure following this approach will be constituted by three steps: the construct identification, the construct measurement and the test validation. This proposal explanation in detail can be seen in Figure 2.7.

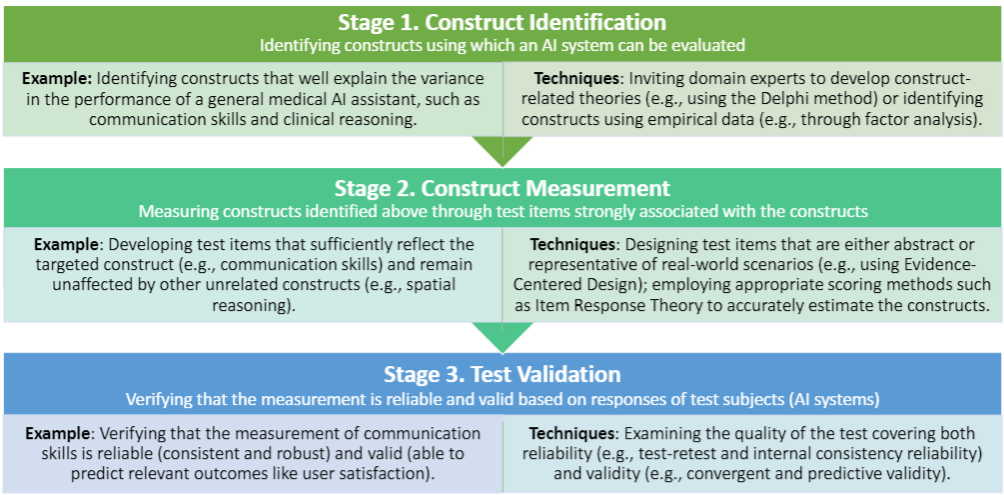

Figure 2.7: Evaluation framework proposal for construct-oriented evaluation grounded in psychometrics by [71].

2.5 Commonsense: the Missing Component in AI

Commonsense reasoning is considered to be one of the areas in which AI has seen very little progress. We could define commonsense reasoning as "the basic ability to perceive, understand, and judge things that are shared by –*are common to*– nearly all people and can reasonably be expected of nearly all people without need for debate." [73]

Advances in AI may have resulted in emergent abilities [72] but they are still narrow and very specialised systems. The key difficulty in modelling commonsense reasoning is that it often "operates" implicitly and does not need to be explicitly articulated, unlike Chain-of-Thought reasoning where every step of the thought process is clearly expressed [19]. (Un)Consciousness about commonsense still remains one of the most interesting features of it, as it is present in many actions and statements. Commonsense is considered the last step and, at the same time, the final and most significative barrier for AI agents to behave human-like and being more general instead of narrowly focused systems.

Many challenges have been introduced to foster the development of commonsense agents. A recent example of this is the DARPA's Machine Common Sense (MCS) initiative, in which the agency proposed the funded initiative for researchers of the "development of a computational model able to mimic the core cognitive capabilities of up to 3-year-old children and a test and evaluation environment for evaluating the models against cognitive development milestones as evidenced in developmental psychology research with children from 0 to 18-months old" [28]. This set of milestones can be seen in Figure 2.8

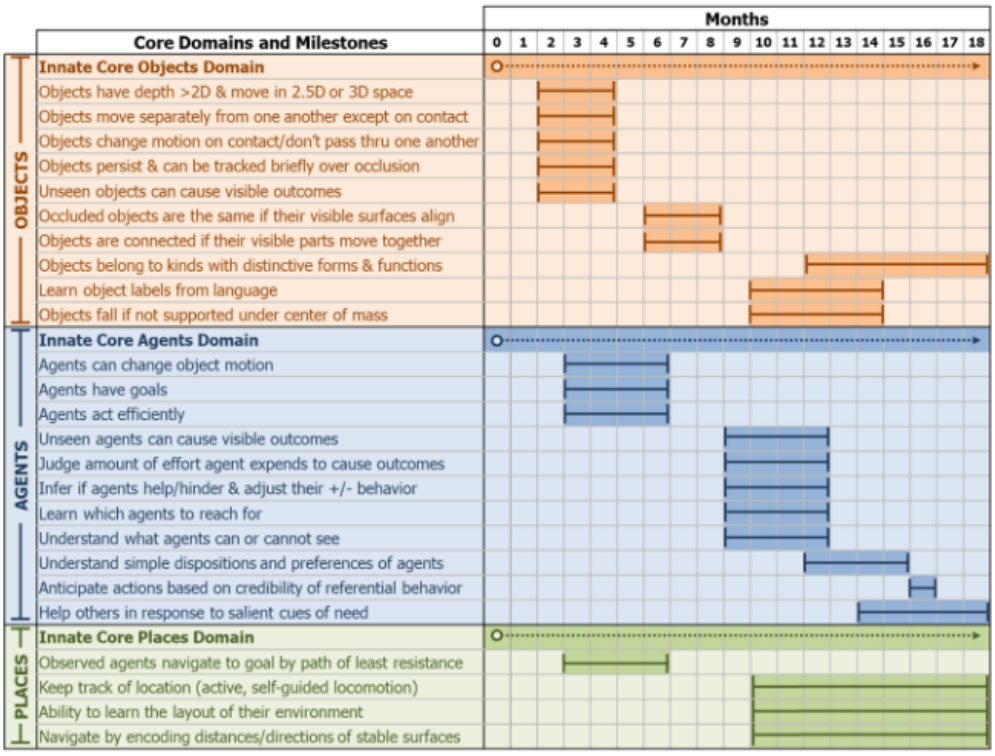

Figure 2.8: MCS proposal of Milestones of Cognitive Development for Children up to 3 years old [28].

They set the foundations for this initiative on the cognitive development Theory of Core Knowledge [63]. This theory states that "human and animal cognition is built upon some structures or systems [...] that allow representing and reasoning about entities of different kinds". Also, it remarks that there are six domains (see Table 2.1) which are considered to be the "cornerstone to set the foundations for future learning". Like the building blocks of human intelligence and commonsense –specially the three first as they correspond to intuitive physics, intentional actors and spatial navigation–.

Domain	Description
Objects	Supports reasoning about objects and the laws of physics
Agents	Supports reasoning about agents that act autonomously to pursue goals
Places	Supports navigation and spatial reasoning around an environment
Number	Supports reasoning about quantity and how many things are present
Geometry	Supports representation of shapes and their affordances
Social World	Supports reasoning about Theory of Mind and social interaction

Table 2.1: Theory of Core Knowledge domains.

In this project, we will take data from some of the agents that were developed in this project and infer their capability profiles through Bayesian triangulation provided by the Measurement Layouts methodology [12].

2.6 Bayesian Modelling and Cognitive Science

Throughout history, Bayesian theory has been applied to many disciplines. Its philosophy of how given some "prior knowledge" it is had about a phenomenon, it can be updated, resulting in "posterior knowledge" thanks to observing the phenomenon and retrieving data about it has been widely adopted. And in the case of cognitive sciences, it has proved its versatility and usefulness in many ways:

1. Using Bayesian statistics to conduct statistical inference based on sampling distributions and null hypothesis significance testing. Basically, helping cognitive science to rigorously analyze its data. [24]

2. A more theoretical approach, in which Bayesian is applied directly as a model for cognitive modelling on trying to "explain how minds make inferences"[16, 41]. This approach is sometimes addressed informally with the metaphor "Bayes in the head".

3. The most recent –and the one this project is based on– pursues "relating models of psychological processes to data"[41, 42]. This is basically taking some prior assumptions about cognition, modelling these assumptions mathematically and evaluate this model against observed behavioural data. This is specially interesting when we model cognitive capabilities as the latent variables of our model, and Bayesian inference allows for determining the capability profile for the individual we have behavioural data.

A specific example of this approach are hierarchical models. These are models which have a set of parameters which characterise a process that generates data –behavioural data in this case– through a likelihood function. A deeper idea of them, adds that the parameters of the model are themselves generated by some other process parameterised by hyper-parameters. This extension of the basic hierarchical model seeks explaining how the parameters that "regulate" the functions that produce data are generated. This is, extending hierarchical model from the theory of task performance, to the theory of the parameters from the variables that control task performance, also called psychological variables in this field –which can be cognitive abilities– [41]. The basic hierarchical model can be seen in Figure 2.9.

In our case, we are interested in how this extension can be applied to model how multiple cognitive processes can be combined to produce observed performance data. An

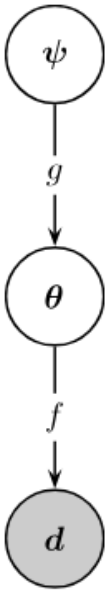

Figure 2.9: A general structure for the hierarchical dependence of basic data-generating process f parameterised by ϕ upon a more abstract process g parameterised by ψ. [41]

example of this model can be seen in Figure 2.10. We will discuss measurement layouts in detail in Section 3.3 but they basically take this idea a step further, linking how these cognitive capabilities, given the cognitive demands an given instance, can model agents' task performance at the instance level.

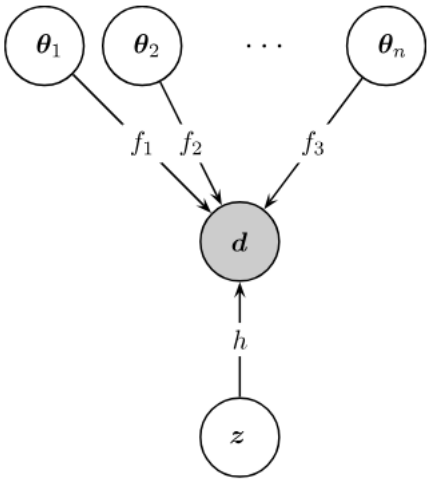

Figure 2.10: A hierarchical modelling approach extension allowing a set of different psychological processes to combine to produce observed data [41]

.

CHAPTER 3
Materials and Methods

3.1 DARPA's Machine Common Sense Program

As it has already been introduced later, this project is demarcated in the context of DARPA's Machine Common Sense Program (MCS). Just as a reminder, the programme sought to address the challenge of machine common sense, two broad strategies are being pursued. Both approaches envision machine common sense as a computational service or a series of machine commonsense services. "The first challenge focused on creating a system that learns from experience, similar to a child, by developing computational models that emulate the core domains of child cognition: objects (intuitive physics), agents (intentional actors), and places (spatial navigation). The second strategy aimed to develop a service that learns from reading the Web, akin to a research librarian, to build a commonsense knowledge repository capable of answering questions about commonsense phenomena in both natural language and images"[1].

This final degree's project is focused on using the Measurement Layouts framework to study the capability profiles from the agents that resulted from the first "challenge". We will enter into more details of how this methodology will serve us for our purpose later on Section 3.3. For training and testing these systems, the Allen Institute for Artificial Intelligence[2] developed a set of tools for simulating scenes for testing AI common sense agents:

- Scene Generator[3]: This ILE –"Interactive Learning Environment"– Scene Generator was used to generate training scenes. This allowed teams participating to train their agents on concepts core to common sense reasoning like physics, occlusion, navigation, localisation, agency, and more. Test scenes were comprised of combinations of these concepts.
- MCS AI2-Thor[4]: This framework was modified to interprets the scene JSON created in the *Scene Generator* to "build the low fidelity 3D environment where teams tested the intelligent system on common sense principles"

As detailed in Section 2.5, the MCS project took as reference the Theory of Core Knowledge [63] and child cognition for guidance on how teams should focus the development of their agents. However, the technical area proposed to develop AI systems able to simulate early-developing, nonverbal common sense focused only on objects, agents, and

[1] https://www.darpa.mil/program/machine-common-sense
[2] https://allenai.org/
[3] https://github.com/NextCenturyCorporation/mcs-scene-generator
[4] https://github.com/allenai/ai2thor

places [27] domains. Therefore, the remaining areas of this theory –numbers, forms and social beings– were not a target for the MCS initiative. Let's introduce how commonsense was intended to be evaluated on agents from this domains perspective while taking the theory of children cognitive development as a reference:

- **Agents Domain**: In the agents domain, commonsense should appear by understanding and interacting with other agents, which include both living organisms and inanimate objects that exhibit goal-directed behaviour. AI systems were evaluated on their ability to distinguish between living and non-living entities, infer the goals behind an agent's actions, and use agents as sources of information.
- **Objects Domain**: Commonsense in the objects domain centered on the understanding of the properties and behaviours of physical objects. AI systems were tested on their grasp of object permanence, the concept that objects continue to exist even when not visible, and their comprehension of physical principles such as solidity and gravity. Tasks evaluated agents ability to track objects moving through space, understand interactions like collisions, and recognise the numerical properties of sets of objects.
- **Places Domain**: In the places domain, commonsense pertains to navigating and understanding spatial environments. AI systems were challenged to keep track of their own location in space, navigate through spaces, and monitor the movement of objects within these spaces. These tasks were designed to see if common sense agents could emulate children's ability to logically deduce the location of objects, navigate environments effectively, and track object movement across different spatial contexts.

Table 3.1 provides an overview of some of the tasks developed per domain.

3.1.1. Agents Evaluations in MCS

The evaluation team was charged with designing studies to assess AI common sense in the above discussed domains. The evaluation team comprised experts in two domains: developmental psychologists with training in the assessment of infant and toddler perception, cognition, and behavior, and software engineers able to program virtual environments in which AI systems could be given tests based on what is known about infants' and toddlers' understandings of objects, agents, and places.

The evaluation had three innovative features: it was motivated by research in developmental psychology, it involved novel hypercube designs, and training data which could be generated by AI development teams using an interactive learning environment (ILE) that was already introduced above.

Research in Developmental Psychology The tests for evaluating systems were motivated by the research literature on early developing competencies, so they were analogous to tasks designed to assess competencies seen in infants and toddlers. Two broad classes of tests were developed: "passive" tasks and "interactive" tasks. Passive tasks were designed to simulate looking-time tasks used with infants, where the duration of looking at various scenes is recorded to infer the child's competencies. Infants typically look longer at unexpected events. Analogous tests for AI involved presenting scenes that appear plausible or implausible to human observers and asking AI systems to generate plausibility scores. Interactive tasks mimic assessments where children retrieve an object, such as a toy hidden in a container. Similar AI tests involved placing AI systems in a

virtual room with a reward object, requiring them to navigate the room and interact with objects to retrieve the reward.

Novel Hypercube Design The evaluations utilised hypercube designs that controlled for numerous variables while manipulating only one variable at a time. Instead of presenting AI systems with a collection of scenes and reporting the overall accuracy, the evaluations used carefully controlled stimuli to assess the effects of specific variables on the AI's common sense reasoning. This is exactly the spirit of the paradigm shift in AI evaluation we presented earlier in Section 2.3. We can have a look at an example of a hypercube for the agent domain task "Agent Identification" in Figure 3.1. Each cell from the hypercube is "assigned" to each kind of experiment of this task and represents a set of crucial characteristics of the item, providing insight beyond whether or not an AI succeeds at that instance.

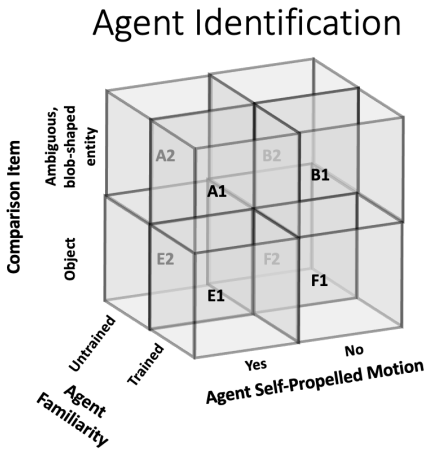

Figure 3.1: Agent Identification task Hypercube, taken from MCS Project Website.

Three teams participated in the first strategy/challenge from the MCS project, each of them developing an agent with the framework provided by MCS. The first team, composed by IBM, and the universities of MIT, Stanford and Harvard presented "CORA"; UC Berkeley, CMU, University of Michigan, the MIT and UIUC developed "MESS" (Model-Building, Exploratory, Social System); and the last team, constituted by Oregon State University, New York University and the University of Utah introduced their agent "OPICS" (Obvious Plans and Inferences for Common Sense).

This project had a duration several years and during this period, up to 7 evaluation "acts" were carried out periodically, allowing agents to be re-trained after receiving feedback from the evaluation process. In each of these evaluations, different commonsense concepts were object of evaluation with some proposed tests –tasks from the domains we introduced above– and metrics for assessing them. Some of the tasks that were used per domain can be seen in Table 3.1.

For the present project, we take performance data from the agents which was generated at evaluations 6 and 7. The process of evaluation generated instance-level results for the aforementioned agents, in which we do not only know whether the systems succeeded, but also get access to details from the specific scene/instance. However, for the specific setting of the Measurement Layouts we were going to use to infer the agents' capability profiles, we were only interested in knowing the cognitive demands –the cognitive demand of a task, is related to the complexity of the task/problem from the perspective of the cognitive ability assessed[66]– each of the instances had. To achieve so, MCS Evaluation team annotated each instance with 9 macro-level variables, representing

Domain	Task Name	Evaluation Procedure	Common Sense Concept assessed
Agents	Spatial Reference	Test whether agents use spatial reference information from agents only, not from objects	Agents can provide solutions to problems and convey knowledge
Agents	Imitation	Test if agents are able to solve a simple problem after viewing an agent model a non-obvious solution (such as touching targets in a specific sequence), demonstrating recognition of the potential value of imitation	Same concept as Spatial Reference
Places	Holes	It is tested if systems can navigate to either a target or an agent that holds the target in a room with holes in the floor that obstruct the AI's path.	Agents can navigate to a target, avoid places that are dangerous, update their location relative to the environment, select the most efficient route, and identify another agent that may have the target
Places	Occluder	Search in a room to obtain a target object that may be invisible behind an occluder, using depth relations to infer possible locations of the target object	Objects exist in 3D space, and persist, even when occluded
Places	Shell Game	Agents are required to track a target object that has been placed in one of several containers either before or after the container is moved	Objects can be tracked over spatial displacement
Places	Spatial Elimination	AI systems must determine where an occluded object must be located given that only one of two occluders in the room is big enough to fully occlude the object	Objects can be located in space by a logical process of elimination
Objects (Passive Task)	Collisions	Provide a plausibility rating for scenes in which a collision may or may not have occurred	One object can be launched into motion when it is hit by another object
Objects (Passive Task)	Object Permanence	Passive recognition that objects do not appear or disappear behind occluders	Objects persist, even when occluded
Objects (Active Task)	Symmetrical Tool Use	Use a simply symmetrical object as a tool, to push or maneuver a target object so that it becomes accessible	Object functions can be predicted by their forms
Objects (Active Task)	Moving Target Prediction	Anticipate the location of a moving object and proceed to that location in order to intercept the object	Objects have trajectories that can be anticipated

Table 3.1: Some Evaluation Tasks from MCS Program per Common Sense Domain.

"generic" cognitive abilities and other high-level relevant properties/features that define the instance and may affect performance; non-ability variables, representing which kind of task the instance was –passive, interactive or a peripheral instance in which the scene featured reasoning–; and 38 micro-level variables representing more specific abilities that were tested in a given instance or other more detailed properties that may be relevant to understand agent's behaviour at the given instance.

For characterising the capabilities of OPICS, CORA and MESS we focused on the macro-level variables. These variables were binary, indicating whether that demand is present or not. Nonetheless, annotation of the macro-level variables represents in a certain way an aggregation of the observed micro-level variables. Therefore, the annotation of micro-level variables was crucial to finally define the task characterisation –"a set of observable, usually constructed, meta-features, expressing cognitive demands and other high-level properties of the task" [12]. This micro-level variables were based on some qualitative patterns or observations from the scene/instance. The following table provides some of these micro-level variables and the qualitative characteristics of scenes that required of the ability they represent:

Ability/Micro-level variable	**Qualitative feature present in the scene/instance**
Reason about motion change at contact between two objects	An object begins to move or changes the direction of movement when it is hit by another moving object; movement by placers doesn't count and features of the room (platform, floor) are not untethered objects
Reason about a target that is not immediately visible	At the start of scene, the target is not within the systems line of sight and it does not see the object hidden or coming out of a popper or moving or on a placer, i.e. it is not visible before the AI starts to act
Reason about object trajectories	The AI needs to anticipate a –visible or invisible– trajectory of a moving object or an object you are about to move –e.g., using a tool to move the target– to obtain the target, even if it is falling down; does not count placer placements
Reason about agents providing solutions to problems	Agents can have targets or can indicate (by pointing) where the target is
Reason about varying number of task-relevant objects	In the instance design some element of the scenes are manipulated.
Obtain a target after forced rotation	The AI is forced to rotate (either 360 in place) or around a cog during the scene
Understand that agents only know about what they have seen	In the instance, agents can see (or not see) where the target is hidden; agents only know what they have seen or experienced

Table 3.2: Some of the micro-level variables used for annotating MCS Evaluation Data.

On the other side, the macro-level variables were: moving object reasoning, core object reasoning, quantity reasoning, agent reasoning, AI reorientation, object permanence reasoning, generalising, tool use and challenging navigation.

3.2 Hierarchical Bayesian Networks and Approximate Inference

Measurement Layouts are semantically-rich specialised Hierarchical Bayesian Networks. To understand them, we have to look at Bayesian Networks first.

3.2.1. Bayesian Networks

Bayesian Networks (BN) benefit from the probability theory, graph theory and statistics. They are also known as *belief networks*, belonging to the family of graphical models. These graphs allow for representing uncertainty in many domains. In Bayesian Networks, nodes represent random variables, and the edges, the probabilistic relationship/dependence between the nodes that it connects. Therefore, due to that Bayesian Networks are usually directed acyclic graphs; the whole structure is a representation of the joint probability distribution over the nodes –random variables [38]. The dependency of variables is given by the direction of the edges. If a variable X_i is a parent node of variable X_j, it indicates that the latter is conditionally dependant on the former.

More precisely, the absence of edge represents a conditional independency. If nodes are assigned a number in topological order, and then we connect them such that each node is conditionally independent of all its predecessors given its parents –this is called ordered Markov property– [48] like this:

$$X_i \perp \mathbf{X}_{\text{pred}(i)\setminus \text{pa}(i)} \mid \mathbf{X}_{\text{pa}(i)} \tag{3.1}$$

In which pa(i) represent the parents of node i, and pred(i) are the predecessors of node i in the ordering. Using this property, we can represent the joint probability distribution (JPD) as:

$$P(\mathbf{X}_{1:N_G}) = \prod_{i=1}^{N_G} P(X_i \mid \mathbf{X}_{\text{pa}(i)}) \tag{3.2}$$

This expression allow us to define easily the joint distribution in a factored form, what eases evaluating possible inferences by applying marginalisation. Borrowing the example provided in [38], we have the Bayesian Network of Figure 3.2, composed by discrete binary random variables.

"It considers a person who might suffer from a back injury, an event represented by the variable Back (denoted by B). Such an injury can cause a backache, an event represented by the variable Ache (denoted by A). The back injury might result from a wrong sport activity, represented by the variable Sport (denoted by S) or from new uncomfortable chairs installed at the person's office, represented by the variable Chair (denoted by C). In the latter case, it is reasonable to assume that a coworker will suffer and report a similar backache syndrome, an event represented by the variable Worker (denoted by W)".

In Bayesian Networks, there are two types of inference *top-down*, also known as predictive inference and *bottom-up*, also called diagnostic inference. Taking our example of Bayesian Network, a case of top-down inference would be: *given that we have observed that an individual suffers from backache, which are the probabilities of uncomfortable to have been installed at the office?*

Using Bayes' Rule, that inference is expressed in the following way:

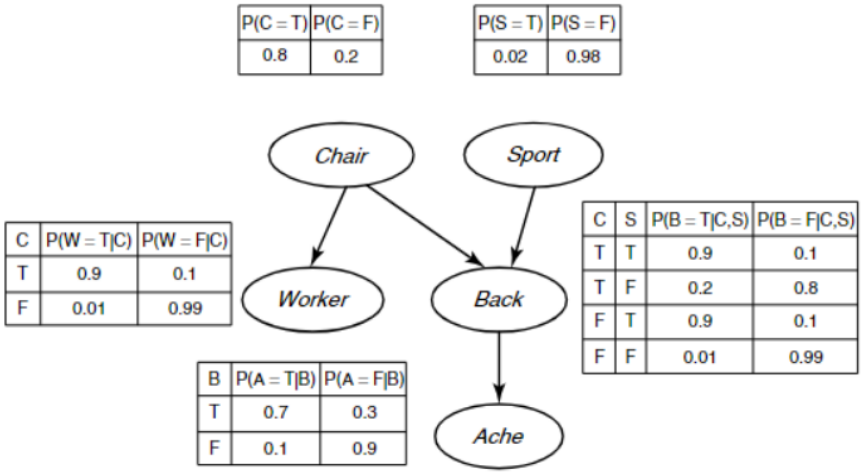

Figure 3.2: A simple Bayesian Networks modelling dependencies between discrete binary variables [38].

$$P(C = T|A = T) = \frac{P(C = T, A = T)}{P(A = T)} \tag{3.3}$$

now we use the JPD expression given by Equation 3.2

$$P(C = T, A = T) = \sum_{S,W,B \in \{T,F\}} P(C = T)\, P(S)\, P(W|C = T) P(B|S, C = \mathrm{T})\, P(A = T|B) \tag{3.4}$$

and

$$P(A = T) = \sum_{S,W,B,C \in \{T,F\}} P(C) P(S) P(W|C) P(B|S, C) P(A = T|B) \tag{3.5}$$

Computing the JPD takes exponential time as it has size $O(2^n)$, where n is the number of nodes. Summing –or integrating when using continuous variables– over the variables is called exact inference, which is known to be a *NP-hard problem*. Given its computational cost, most of the times *approximate inference* is used instead, an inferential method we will delve into later.

3.2.2. Hierarchical Bayesian Networks

As we briefly introduced in Section 2.6, advanced Bayesian hierarchical models enable the explanation of how parameters governing the functions that produce data are generated. Now, let's delve into a more formal definition of Hierarchical Bayesian Networks.

These models serve as a solution for scenarios involving multiple related datasets, where some aspects or features are shared across datasets while others are specific to each. For example, in cognitive science, memory can be studied across various modalities such as verbal, visual, and spatial memory. Each dataset captures unique aspects specific to its modality –idiosyncratic features–, yet they all share underlying processes related to encoding, storage, and retrieval of information –shared features.

To address such complexities, Hierarchical Bayesian Networks (HBN) introduce latent variables that represent unobserved factors influencing the observed data across

datasets. These latent variables accommodate both shared and idiosyncratic effects by allowing parameters to vary at different levels of the hierarchical structure, while coming from the same prior distribution.

If we have J datasets –depending on the problem, j could represent an individual, an experiment, etc.– with N_j data points:

$$D_j = \{(x_n^j, y_n^j) : n = 1 : N_j\} \tag{3.6}$$

The first two options that may come to mind could be: fitting a model per each dataset –i.e $p(y|x; D_j)$ the posterior distribution of the response variable–, what may result in overfitting; or train a single model taking all datasets together –i.e $p(y|x; D = \cup_{j=1}^{J} D_j)$, a choice that might result in underfitting. However, we can use hierarchical Bayesian model in which each dataset (group) has its own parameters θ^j, but they have a shared prior $p(\theta^0)$. This results in a model with posterior distribution:

$$p(\theta^{0:J}, \mathcal{D}) = p(\theta^0) \prod_{j=1}^{J} \left[p(\theta^j \mid \theta^0) \prod_{n=1}^{N_j} p(y_n^j \mid x_n^j, \theta^j) \right] \tag{3.7}$$

Example of Hierarchical Bayesian Model: *Radon* Regression

This is an illustrative example inspired in Chapter 15 from [48]. In this case, it is proposed a hierarchical model to predict the churn rates –the rate to which customer decide to unsubscribe– from a streaming company like Netflix based on a categorical that indicates the region to which a customer/household belongs to, and a binary variable, representing whether the household has youngsters at home or not. It is used a dataset of many households for J regions, in which each j represents a region from a given country. Then, the hierarchical model will fit a regression model for each region, where the parameters of the regression model –slope and baseline–, together with the parameters that model them are the latent variables of the hierarchical model. The graphical representation of the model can be seen in Figure 3.3:

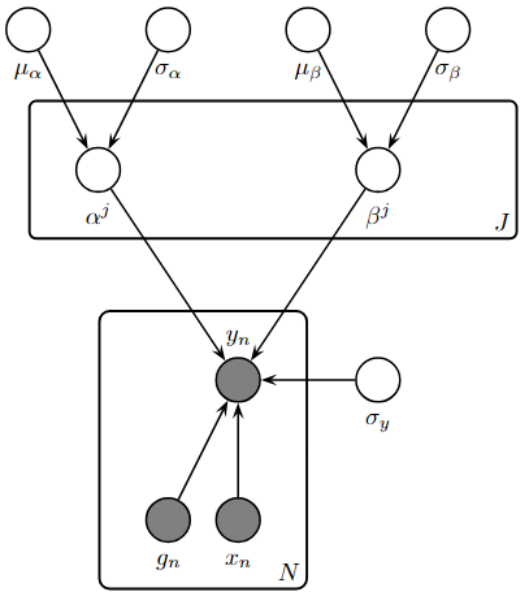

Figure 3.3: Representation of the Hierarchical Model for the Netflix churn rate prediction.

In this model, given the priors assumed that will be presented later, the likelihood of the target variable –log radon levels at a given house from a region– follows a normal distribution like this:

$$p(y_n \mid x_n, g_n = j, \theta) = N(y_n \mid \alpha_j + \beta_j x_n, \sigma_y^2) \tag{3.8}$$

We define hierarchical priors that model the parameters of the regression model for each region:

- For the region intercept. I.e. the baseline churn rate for region j: $\alpha_j \sim \mathcal{N}(\mu_\alpha, \sigma_\alpha^2)$
- For the region slope. I.e. the effect of having youngsters on churn rate for region j: $\beta_j \sim \mathcal{N}(\mu_\beta, \sigma_\beta^2)$.

And the following weak priors for the rest of the parameters/variables:

- $\mu_\alpha \sim \mathcal{N}(0,1)$, $\mu_\beta \sim \mathcal{N}(0,1)$, $\sigma_\alpha \sim \mathcal{C}^+(1)$, $\sigma_\beta \sim \mathcal{C}^+(1)$, $\sigma_y \sim \mathcal{C}^+(1)$

Table 3.3 provides us an explanation of the variables/parameters that appear in the model:

Parameter	Description	Meaning
α_j	Region-specific intercept	Baseline churn rate for Region j
β_j	Region-specific slope	Effect of having youngsters at home for Region j
μ_α	Mean of Region intercepts	Overall mean of α_j across all regions
σ_α	Std. dev. of Region intercepts	Variation of α_j across regions
μ_β	Mean of Region slopes	Overall mean of β_j across all regions
σ_β	Std. dev. of Region slopes	Variation of β_j across regions
σ_y	Measurement noise std. dev.	Noise in the churn rate measurements
y_n	Observed churn rate	Churn rate for house n
x_n	Youngster at home indicator	0 if not, 1 if yes
g_n	Region indicator	Indicates that house n is in Region j

Table 3.3: Summary of Parameters and their Meanings in the Hierarchical Bayesian Model for Churn Rate Prediction

Before moving into the final step of fitting the model and how we can use it to do predictions, let's review what has been defined until now:

1. It has been defined priors for our latent variables: μ_α, σ_α, μ_β, σ_β, σ_y
2. With this latent variables, we have defined hierarchical priors for the latent variables that parametrise the regression model: α_j and β_j.
3. Given these priors, and the dependencies the hierarchical model represents with respect to the target variable, we have the likelihood of our model defined in Equation 3.8.
4. Then, taking Bayes' Theorem, we know that the posterior distribution will be proportional to –in the sense of the "shape"– the product of the likelihood and priors:

   $$p(\theta|D) \propto p(D|\theta)p(\theta) \tag{3.9}$$

   Substituting priors –remember we have two types of priors, the hyperparameters ones and the hierarchical priors– and likelihoods in Equation 3.9 we have:

   $$\begin{aligned} &\mathrm{p}(\mu_\alpha, \sigma_\alpha, \mu_\beta, \sigma_\beta, \alpha_{1:J}, \beta_{1:J} \mid \mathcal{D}) = p(\mu_\alpha, \sigma_\alpha, \mu_\beta, \sigma_\beta) \\ &\times \prod_{j=1}^{J} \left[p(\alpha_j \mid \mu_\alpha, \sigma_\alpha) p(\beta_j \mid \mu_\beta, \sigma_\beta) \prod_{n=1}^{N_j} p(y_n^j \mid x_n^j, \alpha_j, \beta_j, \sigma_y) \right] \end{aligned}$$

5. Now, given the data we have, we should analytically marginalise over the region parameters –α and β– and then, over the hyperparameters to infer the posterior. In this case, this analytical process is really complex, and given that some priors –the ones that are define with the half-Cauchy– are not conjugate to the likelihood, the *exact inference* of the posterior is intractable. However, we can use approximate methods to infer the posterior. To proceed, we present Markov Chain Monte Carlo (MCMC), a numerical approach to approximate robustly the real posterior.

In the next section, we introduce this stochastic technique to approach complex numerical integration problems, and that is used in the Measurement Layouts for approximating complex posterior distributions.

3.2.3. Approximate Inference: Markov Chain Monte Carlo Sampling

Markov Chain Monte Carlo algorithm has been widely adopted in many fields. In the present project, it presents special interest because it application on approximating intractable integration problems in Bayesian statistics like normalisation, marginalisation or expectation.

Monte Carlo integration

The intuition behind MCMC relies on the Monte Carlo integration. This is often used when we want to compute the expected value of a given function of a variable $\mathbb{E}[f(\mathbf{X})]$. This is equivalent to the following integral:

$$\mathbb{E}[f(\mathbf{X})] = \int f(x)p(x)dx \tag{3.10}$$

Where p(x) is the target distribution of $\mathbf{X}$ –in many cases, like in the Measurement Layout, $p(x)$ can be a posterior distribution $p(x|y)$ instead. Solving this problem analytically by numerical integration might be unfeasible as the number of dimensions of $\mathbf{X}$ increases. Then, the idea of Monte Carlo integration is drawing n random samples so that $x_n \sim p(x)$ and to approximate this expected value we take the arithmetic mean of these drawn samples evaluated at the function:

$$\mathbb{E}[f(\mathbf{X})] \approx \frac{1}{N}\sum_{n=1}^{N} f(x_n) \tag{3.11}$$

One key aspect of this approach is that there is no need to draw samples from the whole variable space, but only in which the probability of the sample is significantly greater than zero.

One challenge of this process that it will not be delved into are the wide range of methods to generate random samples from the chosen distribution. However, some of the most used techniques for univariate distributions are *cumulative distribution function inverse sampling*, *rejection sampling* or *importance sampling*. These methods serve as the building blocks for sampling from more complex distributions like multivariate ones.

The MCMC Algorithm

The basic idea of the MCMC methods is to generate a Markov Chain –a stochastic process where the probability of transitioning to any future state depends solely on the present

state [48]– whose stationary distribution matches a target density function $p^*(\theta)$. In the present context, this target density will be a posterior $p^*(\theta) \propto p(\theta|D)$ of some parameters.

With this technique, the random walk generated induces that the "fraction of time" spent across the values from the state space –parameter space– is proportional to the $p^*(\theta)$ it is desired to be estimated. The connection between Markov Chains and the Monte Carlo methods is that by "drawing correlated –correlated because of the Markov Chain definition– samples from the chain we can perform Monte Carlo integration with respect to p^*".

The simplest version of MCMC is the Metropolis-Hastings algorithm, the first MCMC algorithm, proposed back in 1953 [47]. The pseudocode for the algorithm can be seen below 3.1.

Algorithm 3.1 Metropolis-Hastings MCMC Algorithm

Initialise x^0

for $s = 0, 1, 2, \ldots$ **do**

Define $x = x^s$

Sample $x' \sim q(x'|x)$

Compute acceptance probability

$$\alpha = \frac{\tilde{p}(x')q(x|x')}{\tilde{p}(x)q(x'|x)}$$

Compute $A = \min(1, \alpha)$

Sample $u \sim U(0, 1)$

Set new sample to

$$x^{s+1} = \begin{cases} x' & \text{if } u \leq A \text{ (accept)} \\ x^s & \text{if } u > A \text{ (reject)} \end{cases}$$

Let's briefly explain the algorithm. A initial estimation for the variable it is desired to be estimated is chosen (x_0), then, until the desired number of samples are drawn the following steps are executed:

1. It has been defined a *proposal distribution* q[5] which given the current state –the last sample draw– of the random walk, proposes a new state x' to move with probability $q(x'|x)$.

2. Then, we compute the probability of given the current state, to accept the proposal –this is called the acceptance probability "A". Usually this probability would be minimum between 1 and the ratio –α in the algorithm– of $p^*(x')$ with respect to $p^*(x)$, but to avoid the proposal favouring certain states/values it is introduced the Hastings correction, which is the form that can be seen in the algorithm.

   Note that $\tilde{p}$ represents the unnormalised form of p^* –i.e. $p^* = \frac{1}{Z}\tilde{p}(x)$, where Z is the normalisation constant. This estimation assuming it is being approximated a posterior, is usually in the form of the product of its likelihood and prior.

[5]There are many approaches for choosing a "valid" proposal, but when $p^*(x)$ is a posterior, very often it is opted to include observed data when conditioning the sampling –i.e. $q(x'|x)$ now is $q(x'|x, D)$. This approach is called data-driven MCMC [48]

3. After it, if acceptance probability is higher than 0.5, it does not necessarily imply that the new sample is accepted as the current state. To proceed, it is sampled a value "u" from a uniform distribution $U(0,1)$, which is compared with the acceptance probability "A". If "A" is greater than "u", the proposed sample becomes the new state, otherwise, the current state is taken again as the new state and the proposal distribution samples a new proposal state.

   After "s" steps –i.e. the chain is composed by "s" states–, the estimation for the parameter is trying to be inferred is the sample mean from the values that compose the chain.[6]

This was the basic form of the MCMC algorithm, and, as many other variants of it, struggles when sampling at high dimensional spaces, as it basically relies on random search based on local perturbations of the current state. In this sense, it was proposed Hamiltonian Monte Carlo (HMC) [23], which leverages concepts from Hamiltonian mechanics for defining a more "informed" sampling chain that introduces gradient information. This is the *approximate inference* algorithm that we use in the Measurement Layouts. First, let's introduce a few concepts from Hamiltonian mechanics which are crucial to understand the algorithm:

1. The motion of a particle is characterised by its position q and its momentum p. The combination of position and momentum is called *phase space*. The energy of the particle is given by the Hamiltonian function, which depends on its potential energy $\mathcal{E}(q)$ and its kinetic energy $\mathcal{K}(p)$:

$$\mathcal{H}(p,q) = \mathcal{E}(q) + \mathcal{K}(p) \tag{3.12}$$

2. When transferring these concepts to Bayesian statistics, these terms are redefined in the following way:

$$\mathcal{E}(q) = -log\tilde{p}(q) \tag{3.13}$$

   Where $\tilde{p}(q)$ is the unnormalised distribution from the posterior of the parameter we want to approximate (q).

$$\mathcal{K}(p) = \frac{1}{2}p^T\Sigma^{-1}p \tag{3.14}$$

   Where Σ is the inverse mass matrix. The choice of this positive definite matrix is relevant. The most common approach consists on setting it to the identity matrix for the burn-in sampling step, and then computing the empirical covariance matrix using the sampled values for the parameter that is being estimated like this:

$$\Sigma = \frac{1}{N}\sum_{i=1}^{N}(q_i - \bar{q})(q_i - \bar{q})^T \tag{3.15}$$

   where q_i are the samples, $\bar{q}$ the mean of the samples, and N is the number of samples collected after the burn-in period.

Below we have the pseudocode for the HMC Algorithm. We will explain the variation that uses the "leapfrog integrator", other well-known versions are the Euler's and its modified version, which do not even need keeping the momentum parameter.

Now, let's briefly break it down step by step:

[6] It must be noted that usually the first samples from the chain called mixing time or burn-in time samples, which are discarded because they are used to converge to the target distribution.

Algorithm 3.2 Hamiltonian Monte Carlo Algorithm

for $t = 1 : T$ **do**
  Generate random momentum $v_{t-1} \sim \mathcal{N}(0, \Sigma)$
  Set $(q'_0, p'_0) = (q_{t-1}, p_{t-1})$
  Half step for momentum: $p'_{1/2} = p'_0 - \frac{\eta}{2}\nabla\mathcal{E}(q'_0)$
  for $l = 1 : L - 1$ **do**
    $q'_l = q'_{l-1} + \eta\Sigma^{-1}p'_{l-1/2}$
    $p'_{l+1/2} = p'_{l-1/2} - \eta\nabla\mathcal{E}(q'_l)$
  end for
  Full step for location: $q'_L = q'_{L-1} + \eta\Sigma^{-1}p'_{L-1/2}$
  Half step for momentum: $p'_L = p'_{L-1/2} - \frac{\eta}{2}\nabla\mathcal{E}(q'_L)$
  Compute proposal $(q^*, p^*) = (q'_L, p'_L)$
  Compute $\alpha = \min\left(1, \exp\left[-\mathcal{H}(q^*, p^*) + \mathcal{H}(q_{t-1}, p_{t-1})\right]\right)$
  Set $q_t = q^*$ with probability α, otherwise $q_t = q_{t-1}$

1. For starting the algorithm, it is taken a random initialisation of the position –the parameters to be approximated. Then, we obtain a random value for the momentum coming from a normal distribution with mean 0 and using the mass matrix as its covariance matrix. This will be our initial conditions for this t step.

2. It is computed the gradient of the potential energy, and it is performed half step of the momentum update. The update is performed using the gradient of the potential energy weighted by its step size, represented by the parameter η.

3. It is performed L leapfrog steps to update the initial position and the (half-)updated momentum. The momentum is updated following the same procedure as detailed in the previous step, while the position is updated weighting –pre-multiplying– the (half-)updated momentum by the inverse mass matrix and the step size. This is repeated for $L - 1$ leapfrog steps.

4. It is performed the final update for this iteration of the location/position q and the rest of the half-step for the momentum p. This two last updates compose the new proposed state $(q^*, p^*) = (q'_L, p'_L)$

5. We compute the acceptance probability of this new phase space, given by the expression : $\min\left(1, \exp\left[-\mathcal{H}(q^*, p^*) + \mathcal{H}(q_{t-1}, p_{t-1})\right]\right)$. The reason for this expression of the acceptance probability arises from the fact that it we have chosen appropriate priors and we have modelled Hamiltonian mechanics appropriately, the process should be energy conserving and then, the differences between the Hamiltonian/energy function between the proposal and the previous state would be zero. Therefore, obligating to take the new state.

In our methodology for approximate inference, we use HMC with no-U-turn sampler [36], which chooses the number of leapfrog steps L to be large enough that the algorithm explores the states that keep constant energy without the need to stay in the same position.

3.3 Measurement Layouts

As it will be recalled many times later and has already been stated out before, "a Measurement Layouts is a specialised, semantically-rich version of Hierarchical Bayesian Networks, which allow us to model how task-instance features interact with AI systems capabilities to affect performance".

From the cognitive science perspective, it can be understood that the performance at a given task from a cognitive system –in essence, a member from the machine kingdom–, is a function of that task demands and the capability levels. The relation between the two depends on the characteristics of the task and the capabilities being evaluated. For example, in a language comprehension task, the complexity of the sentences to be understood requires a comprehension capability that matches or exceeds the complexity level of the sentences. Another less abstract example, and introduced in [12] states: "in a memory problem, the number of objects to remember in a particular task demands a memory capability level at least as high as the number of objects".

It must be taken also into account the fact that simply observing the differences of the capability with respect to the demand does not explain all performance variance. Other unaccounted factors, noise and cognitive biases could have its influence on it.

Now, we will explain some key terms for understanding this framework from a high-level perspective and see how they are "connected" through the Measurement Layout, and then, we will dive into formalising them.

Measurement Layout: Some definitions

- **Cognitive Task**: Taking the definition from [32], a cognitive task is an "interactive series of stimuli that allows for different observable behaviours on the subject and it is cognitive as far as performance is involved, and its interface can be altered or simulated without affecting the nature of the process".

- **Task Characterisation**: A set of meta-features that can affect performance in a given task. When characterising cognitive tasks, this may include cognitive demands and other high-level properties of the task.

- **Cognitive Ability**: We have already defined this concept a few time previously, but a more precise definition provided in [32] is: "a gradient property of an interactive system in the machine kingdom that allows the system to perform well in a class of cognitive tasks". Cognitive abilities are inherently tied to an organism's capacity and cognitive resources. Any scale measuring cognitive abilities reflects that they are gradient features, meaning a higher magnitude indicates greater capability.

  When a system/individual has a singular behaviour in a class of task in terms of only being able to carry out very specific and maybe considerably complex settings of it, it is preferable to call this attainment instead of ability.

- **Cognitive Profile**: It derives from the concept of psychometric profile, which is defined as the set of behavioural features which are measured for a particular individual/system. But in our applied case, it has a more precise definition, being the following triplet $\langle C, B, R \rangle$. These three elements are vectors of capability levels, bias and robustness respectively.

  The former represents what the agent can do; bias values, limitations or preferences that may affect performance in a "less monotonic way" [12]; and the latter accounts

for reliability issues, i.e., unexplained or random effects (noise) on either the agent or the environment.

- **Compensatory Capabilities** : In this context, compensatory behaviour is observed in an agent when it lacks a particular capability but has developed another ability to such an extent that it can compensate for the weaker one in a specific task. This concept is linked to whether the performance of an agent at a given instance requires all the capabilities from its cognitive profile to some extent, or conversely, whether some capabilities can compensate for the absence of others. The first step to determine whether the capabilities are compensatory is to model if they are independent or not.

- **Measurement Layout**: The measurement layouts is a directed acyclic graph based on the concept of Hierarchical Bayesian Networks that connects through "linking functions" the meta-features coming from the task characterisation with the cognitive profile of an agent in such a way that allows predicting the system's performance.

  Measurement layouts are Hierarchical Bayesian Networks in which nodes represent every meta-feature from the instance or an element from the system's cognitive profile. These nodes are the roots of the HBN, and the connection between them represents conditional dependencies, as they do in normal hierarchical models. The relationship between dependent nodes has not only a probabilistic interpretation, but a semantic one. In this sense, when saying semantic it is referred to that they encode domain-knowledge.

  The set of all the nodes and their dependencies, as expressed by the Equation 3.2, encodes a probability distribution, but we will delve into more details about formalising the measurement layouts in Section 3.3

  Now let's study specific types of nodes and "components" we can find in measurement layouts that will allow us to understand the "link" between the aforementioned concepts.

  - **meta-features**: They come from the task characterisation and are fixed observable values.
  - **Cognitive Profile Nodes**: Elements from the cognitive profile that are relevant to the assessed task's demands. These may include capabilities and biases. Cognitive profile nodes can combine with meta-features and feed into derived nodes. As explained in Section 2.6, when trying to model complex cognitive mechanisms, we define prior distributions for the parameters that regulate the variables –in this case, these variables are the elements from the cognitive profile– that are in control/influence task performance.
  - **Linking Function**: A mathematical expression that maps values from the output of one node's probability distribution to the input of the another. Some examples that are actually used in some topologies of the measurement layouts are: the sigmoid function of the difference between capability and demands; a product of capabilities when they are not compensatory, etc.
  - **Derived Nodes**: They are instance-level inferences. They surge as a combination of meta-features and elements from the cognitive profile, and the parameters of the probability distribution that model them derive from the nodes they depend on. Indeed, due to measurement layouts encoding domain-knowledge information, their parameters are computed through the linking functions, and their outputs are used as the summary statistics of their distribution.

> A relevant example of derived node that is present in every topology is the observable performance. However, the most frequent derived nodes are the "Intermediate Non-Obervable Nodes" (INON), which represent intermediate non-observable performance or effects.

It is important to insist on how relevant is domain-knowledge and having a deep understanding of how cognitive capabilities and the available meta-features from the instance are expected to affect the values of instance-level inference nodes. This can help to build richer connective structures that introduce higher-order and complex relationships between nodes, "with certain INON nodes requiring the nuanced confluence of many cognitive profile elements and meta-features" [12]. This complex structure is not intended solely to enhance predictive power in the model. Rather, it serves the crucial purpose of accurately distinguishing between success, failure, and the nuances of complex behaviours that a well-designed benchmark for evaluating agents must address.

The primary distinction between HBNs and measurement layouts relies in their primary goal: "capturing the hierarchical dependencies between capabilities and demands, rather than encoding information on hyper-priors". Indeed, the process of building the connective structure of the measurement layouts is done by applying ideas from cognitive modelling.

As an example of this, we can look at important decision of determining if some capabilities are independent and, if so, whether they are compensatory, using additive or multiplicative expressions accordingly. If they are not independent, they need to be linked with a mathematical function expressing their dependency.

Given a system's cognitive profile and the characteristics of a new task, performance can be predicted using top-down inference. Before this, the cognitive profile must be inferred from the observed performance on other tasks using bottom-up Bayesian inference. To achieve this process, we employ PyMC's inference engine, which we will introduce in 3.4 and the No U-Turn Sampler –introduced in Section 3.2– approach to Bayesian approximate inference. To accurately predict a system's cognitive profile, the test battery must control for alternative explanations, allowing performance results to "triangulate" latent capabilities.

High-level introduction of the measurement layout

To grasp an actual idea of the functioning of this framework, let's look at an introductory high-level example taken from its original paper [12], which can be seen graphically in Figure 3.4.

This measurement layouts displays the two types of inferences that are used within this framework. This topology illustrates an agent which has been observed to perform badly when exploring a 3D environment to find a reward, what demands understanding that the reward still exists when occluded (object permanence). It requires remembering where it is, and successfully navigating to it. But we do not know if the reason for failure arises from the lack of object permanence, limited memory or navigation skills. This type of task instances are represented by the red –red indicating failute– circle with the number 3.

However, we have found that it performs well when the task only demands complex navigation –green circle with the number 1–, and in tasks which imply both navigation and memory only –green circle with the number 2–. By bottom-up inference, having observed this patterns of behaviour, we could intuitively determine that the agent has decent navigation and memory ability, while it has a bad object permanence ability. This

bottom-up triangulation has allowed us to; given the observed data, to infer the agent's cognitive profile.

On the other hand, we see a new incoming task represented by the blue circle with the number 4. This task is very demanding from the point of view of object permanence, so we can infer that it is very likely that the system fails at this new task. This is top-down inference.

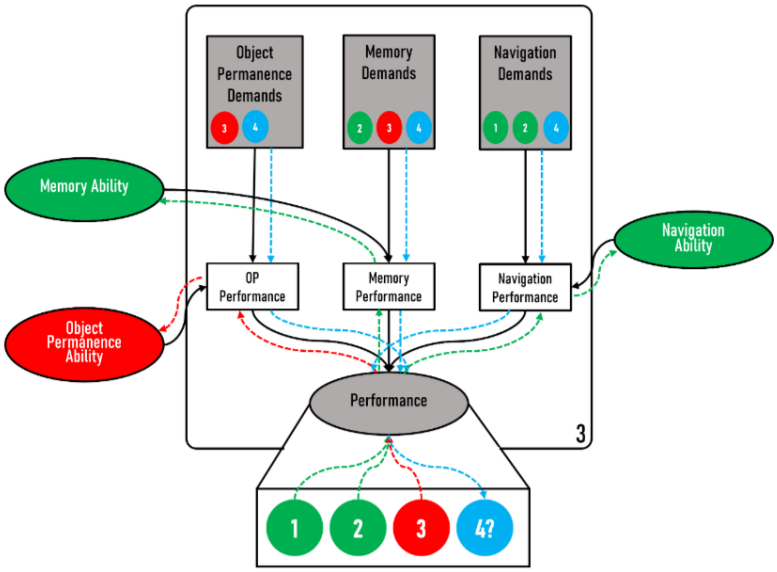

Figure 3.4: Example of Triangulation in the measurement layouts. Bottom-up inference from three tasks (in green and red for success and failure respectively) leading to the cognitive profile. Top-down inference (in blue) predicting failure for the fourth task. Taken from [12].

Formalising the Measurement Layout

We can formally define the Measurement Layouts as a "model that parametrises a probability distribution over the performance of a subject on a given instance, using the demands of the instance and the capabilities of the subject".

Let i denote instance index and j denote subject index. Let $\theta_j \in \mathbf{R}^M$ denote the capabilities of subject j, and $x_i \in \mathbf{R}^M$ denote the demands of instance i. We often consider the margin between the l-th capability and demand, defined as $\theta_{j,l} - x_{i,l}$. The introduction of margins allows to solve a problem introduced in Section 2.4, and it is that the capabilities of a system and the demands of the task should be in the same scale. Also, with an appropriate choice of the priors probability distributions for them, the capabilities can be positively correlated with performance. As we already introduced in the previous section, there may be other elements from a task characterisation which are are not strictly demands and may affect performance without the need to be combined or linked with capabilities through margins; let us denote those by ϕ_j. Also, there may be other parameters which are common to the evaluation setting and are present for all agents and tasks, which we will denote as ξ.

Then, taking these terms and our initial definition of the measurement layout, the probability distribution (likelihood) of performance for the instance i and individual j has the following expression:

$$p(y_{i,j} \mid \theta_j, \phi_j; x_i; \xi) \tag{3.16}$$

Then, given the prior distributions on θ_j, ϕ_j and ξ, we can define the following posterior distribution over all the parameters after having observed n instances for each of the m subjects similarly as we did in 3.7 for Hierarchical Bayesian Networks and using Bayes' Theorem to infer that the posterior is proportional to the product of the likelihood and the priors:

$$\pi\left(\theta_j, \phi_j, \xi \mid y_{1:n,1:m}, x_{1:n}\right) \propto \pi(\xi) \prod_{j=1}^{m} \left[\pi_j(\theta_j, \phi_j) \prod_{i=1}^{n} p(y_{i,j} \mid \theta_j, x_i, \phi_j; \xi) \right] \tag{3.17}$$

This expression "is" the Measurement Layout, and is the posterior probability we aim to approximate using the HMC algorithm provided by the PyMC framework. The approximate inference of this posterior also allows us to estimate the marginals for the parameters for each subject.

Notice that, while the capabilities for each subject are independent on the others given the values of the demands and ξ, the posterior does not factorise for ξ and (θ_j, ϕ_j) due to the presence of all the parameters in the likelihood term. As a result, if a new subject is added, both distributions will change. To avoid this appreciation, it can be fixed a Dirac-delta distribution on ξ: $\pi(\xi) = \delta_{\xi_0}$. This results on a simplification of the posterior of the subjects because it factorises:

$$\pi\left(\theta_j, \phi_j, \xi \mid y_{1:n,1:m}, x_{1:n}\right) \propto \prod_{j=1}^{m} \left[\pi_j(\theta_j, \phi_j) \prod_{i=1}^{n} p(y_{i,j} \mid \theta_j, x_i, \phi_j; \xi) \right] \tag{3.18}$$

This is a really important assumption, as it allows us to infer the posterior per each individual/subject separately, which is more feasible from the computational perspective

The capability parameters ϕ_j acquire meaning from the specific measurement layouts formulation, including their interaction with demands (e.g., via the margin). Therefore, capability posterior distributions from different measurement layouts are not comparable, even if they are compared against the same demands but differ in how margins are combined.

Example: Bernoulli Response and Single Margin and Demand

To give an example of it works, let's consider that performance in a task is a Bernoulli variable and we only have one single demand and a unique margin. Therefore, a possible expression for measuring the success at the task could be:

$$p\left(1 | \theta_j, x_i, \phi_j; \xi\right) = \frac{1}{1 + e^{-\xi(\theta_j - x_i)}} \tag{3.19}$$

Which exploits a logistic function of the margin to compute the probability of success. In this case ξ represents the slope of the logistic, and it is desirable to fix it a ≥ 0 to make sure that larger capabilities correlate with larger probability of success. This slope can be an "inferrable" parameter to be adjusted automatically using the hierarchical measurement layouts formulation from Equation 3.17. However, a general approach used is to fix a value so that the scaled margins have roughly the same orders of magnitude across capabilities, which ensure that the various margins can impact the final probability equally. This is done by setting the following value for the logistic function:

$$\xi = \frac{\log \frac{1-p}{p}}{max\_margin} \tag{3.20}$$

Where p is the probability that wants to be assigned to the maximum margin –represented by max_margin– that can take place for each combination of the capability and demand.

Example: Multiple Margins

This is a more generic setting, in which we may find that there are multiple capabilities that may be combined by linking functions and then compared to a demand, or we simply may find multiple margins that are result of "simple" margins –simple in the sense that comes from the "comparison" of an only demand and capability–.

Going back to the high-level characterisation of the measurement layouts above, these margins, transformed into "sub-probabilities" using Equation 3.19 could represent intermediate effects/performance, so we might be interested on combining them to obtain a second derived node that aggregates this information, or maybe this aggregation may be used to infer the parameters that characterise the final performance probability distribution. Hence, if we have multiple probabilities (σ_l) coming from the expression given by Equation 3.19, we have multiple choices for combining them:

- Product: the product of this probabilities would result in a probability which is smaller than the minimum σ_l. From the cognitive science perspective, this has an actual interpretation that can be explained as each σ_l representing the performance of a sub-task and the performance across them is not-compensatory, so that it is needed to pass all of them to succeed in the general task.
- Complementary product: this option is given by the following expression:

$$1 - \prod_{l=1}^{L}(1 - \sigma_l) \tag{3.21}$$

  This has the opposite interpretation to the product of σ values, and is introduced when the sub-task performance can compensate for others.

There are other options, like using the maximum or minimum σ, using weighted means, but we will principally focus on the first two presented.

Binary demands

When the demands themselves are binary, then we need an revised "margin", as $\theta_j - x_i$ is not easily interpretable. A possibility is:

$$p(y_{i,j}|\theta_j, x_i) = 1 - ((1 - \theta_j)x_i) \tag{3.22}$$

The goal of this margin is to reflect that the demand can be present, in which case the capability is utilised (and the sub probability is θ_j), or the demand can be absent, in which case the outcome of the sub-problem is 1 independently of the capability. This approach is only valid when the capabilities are bounded in [0, 1]. While this may make sense for certain capabilities and binary demands, there are cases in which the capabilities "make more sense" if they are modelled not to be bounded, as it is the case when prior

distributions for them are normal. In these cases, it is proposed the following alternative for the logistic function for computing probability of success at sub-tasks:

$$p(y_{i,j}|\theta_j, x_i) = \sigma(\theta_j - \log(x_i + \epsilon)) \tag{3.23}$$

Where ϵ is just a constant used to ensure numerical stability and the margin is redefined to be $\theta_j - \log(x_i + \epsilon)$. This is interpreted has a similar interpretation to the previous alternative

Introducing Noise in the Measurement Layout

Sometimes it is desirable to introduce a noise component in our measurement layouts formulation as follow:

$$p(1 \mid \theta_j, x_i, \phi_j; \xi) = (1 - \varphi_j) \cdot \tilde{p}(1 \mid \theta_j, x_i, \phi_j; \xi) + \varphi_j \nu_j \tag{3.24}$$

Where the probability of success, obtained by combining capabilities and demands and denoted as $p(1|\theta_j, x_i, \phi_j; \xi)$, is weighted with a constant component $\varphi_j \in [0, 1]$ that is a agent-specific parameter which is used to weight the demand-dependent part and the demand-independent part $\nu_j \in [0, 1]$.

This is used in the following way:

1. Draw a binary random variable with probability φ_j
2. If that sample is 1, use the measurement layouts to predict probability of success; otherwise, the latter is set to ν_j

These last two parameters are not random, and can be approximated with the measurement layout. The motivation for including noise is as follows: if the demands are not predictive for the subject and the specific measurement layouts formulation, the posterior for φ_j will give high weight to values close to 1. In this case, the prediction of success will be the constant ν_j, independently of the demand x_i.

There is another approach for introducing noise, that indeed allows bounding the predicted performance as we decide to, between "a" and "b" as follows:

$$p(1 \mid \theta_j, x_i, \phi_j; \xi) = a_j + (b_j - a_j) \cdot \tilde{p}(1 \mid \theta_j, x_i, \phi_j; \xi) \tag{3.25}$$

This is equivalent to Equation 3.24, but $a_j = \varphi_j \nu_j$ and $b_j a_j = \varphi_j$. It is suggested that a_j is a fixed value, while b_j can be learnt for each subject independently and indicates the level of "reliability" that the measurement layouts has for the considered subject. This captures both the randomness of the subject as well as the unexplained factors in the data. The estimation of this last parameter introduces the following possibilities for interpreting the measurement layouts inferred values for them:

- A subject with a lower estimated capability level might outperform one with a higher estimated capability level for a specific dataset. This happens if the latter has a lower b_j estimated. This would suggest that that the most capable subject may fail in ways not captured by the considered demands, possibly due to randomness or unconsidered demands.
- If a subject has an estimated b_j<1 and we introduce a new demand to explain some of the failures, the other capability estimates will be expected to change less than if b_j was not present.

The importance of the Choice of the Prior Distribution for Capabilities

The choice of the prior distribution for the element of the cognitive profile to be inferred are important for their further interpretation. This concept arises from the psychology's concept of the level of measurement. The level of measurement corresponds to "a classification initially proposed by Stevens [62] in order to describe the nature of information contained within numbers assigned to objects or subjects –abilities in our case. It refers to the degree to which characteristics of the data may be modelled mathematically." [39]

In our case, the choice of prior must be accompanied by a justification of which scale we are trying the abilities to be located at. Table 3.4 provides an overview of some of the scales that have been tried to be modelled within the measurement layout, together with its corresponding measurement property following Stevens' classification and its practical implementation and interpretation.

3.4 PyMC

PyMC [56], is a probabilistic programming language available for Python that allows us to build the measurement layouts, which as we have pointed out many times previously, are "is a specialised, semantically-rich version of Hierarchical Bayesian Networks...". And "that" for what it helps us with: building Bayesian models and fitting them with Markov Chain Monte Carlo (MCMC) methods. In our work, we make use of the No U-Turn Sample (NUTS), an extension of the Hamiltonian MC algorithm that we introduced back in Section 3.2.3. Table 3.5 introduces some of the classes and methods that we have used the most for defining the hierarchical model that the measurement layouts represent. This table also includes use cases of them.

The pseudocode in Algorithm 3.3 provides a general idea of how it was proceeded for defining the measurement layouts using PyMC.

Scale/Level	Measurement Property	Practical Implementation	Interpretation
Ordinal	Comparison, rank order	The choice of prior allows only assigning ranks to abilities based on performance metrics or observed behaviour without assuming equal intervals between ranks.	Allows for determining relative positioning of abilities (e.g., better, worse) but does not quantify the magnitude of difference between ranks.
Interval	Difference, affinity	Define abilities on a scale where the difference between any two values is meaningful and consistent, but the ratio between them is not. The normal distribution fits this scale well, especially for calculating differences between capabilities and demands.	Quantifies the degree of difference between abilities. Measures of central tendency like mode, median, and mean, and measures of dispersion like range and standard deviation have sense for this scale. Ratios of differences can be used, but absolute ratios are not meaningful.
Ratio	Magnitude, amount	Define a prior for abilities such that the scale has a true zero point, where both differences and ratios are meaningful. The lognormal distribution was considered for this scale to try to represent abilities with an absolute zero and proportionality.	Provides the most detailed level of measurement. Allows for all mathematical operations, including ratios. This scale is useful for understanding compensatory capabilities, where combined abilities can meet a total required capabilities. Modelling this kind of scale is more challenging.

Table 3.4: Level of Measurement Theory and its relation to modelling Abilities Scales

Name	Class/Method	Description of Use
Model	Class	This allows us creating the hierarchical model that the measurement layouts represents
MutableData	Class	This allows us to introduce environment variables in the defined model, i.e., the demands of the task
Deterministic	Class	It is used for defining derived nodes
Distributions	Class	This allows us to introduce the capabilities as random variables and assign prior distributions to them
Sample	Method	Applies MCMC sampling using the defined model. In our case, we use the NUTS sampler, an extension of the Hamiltonian MC algorithm. This corresponds to the bottom-up inference we introduced in Section 3.3
Model to Graph	Method	Allows us to obtain a graphical representation of the model
Sample Posterior Predictive	Method	It is used after bottom-up inference, once it has been fitted the distributions for our capabilities, in order to predict hold-out data. This process corresponds to top-down inference

Table 3.5: Classes and Methods from probabilistic programming language PyMC used in the Measurement Layouts

Algorithm 3.3 Setup Measurement Layouts in PyMC

Input: relevantData, taskResults, noise_type, prior, compensate

Output: Model m

Initialize global constants and settings

Initialize ability ranges for different capabilities

procedure SETUPMODEL(trainingData, taskResults, noiseType, prior, compensatory)

$m \leftarrow$ PyMC Model Class

with m:

Define data input variables (meta-features)

Define priors for abilities

Define derived nodes (INON nodes) based on abilities and demands margins or linking functions chosen.

Aggregate performances (if derived nodes representing intermediate performance was computed) into a single performance measure and depending on the **compensatory** setting.

if includeNoiseBeforePerformance **then**

Incorporate noise into performance based on noise type

end if

if binaryOutputs **then**

Define task performance using Bernoulli distribution

else

Define task performance using Beta distribution

end if

Return model m

end procedure

CHAPTER 4
Data Processing and Exploratory Analysis

Data preparation process was already introduced in Section 3.1, but let's do a quick summary of how the evaluation data from the MCS project was generated and preprocessed:

1. Firstly, developers of MCS participating agents OPICS, CORA and MESS used Scene Generator and the AI2-Thor simulator –both introduced in Section 3.1– to create and run their own set of test scenes to train on concepts core to common sense reasoning.

2. After it, the multidisciplinary evaluation team from MCS designed the tasks to assess AI common sense in three domains: Objects, Agents, and Places. We recall to Table 3.1 for a description of some of the tasks used for the evaluations.

3. During the program, several evaluation acts took place. In each of these evaluations, different commonsense concepts were object of evaluation with some proposed tests and metrics for determining the degree of accomplishment. As we already mentioned, we focus on the last two evaluation processes –from now on we will refer to them as Evaluation 6 and Evaluation 7–.

4. The evaluation acts were carried out in such a way that for each instance provided for each agent, a important number of meta-features characterising the behaviour of the agent, the characteristics of the scene and the final results were generated. However, for our specific purpose of deriving systems' cognitive profiles with the measurement layouts, we needed that the instances were annotated with its cognitive demands –as discussed in Section 3.3–, and these were not derived directly from the MCS evaluation process.

5. As discussed in Section 3.1, the MCS Evaluators assisted us by annotating each instance with various variables. These include 9 macro-level variables representing generic cognitive abilities and other high-level features that define the instance and may impact performance, non-ability variables indicating the type of task –whether passive, interactive, or peripheral with reasoning–, and 38 micro-level variables representing specific abilities tested or other detailed properties relevant to understanding the agent's behaviour in each instance.

6. This process generated two datasets –one for each evaluation act– that we used for inferring the agents' capability profiles. Both datasets have the same structure of columns, but they do not contain test results for the same type of tasks, due to that evaluation acts overlapped only in some tasks –the fact that the same task

appears in both evaluations does not necessarily mean that the observed results are the same, due to that the instances might not be repeated from one evaluation to another.

Table 4.1 represents the structure of the dataset –data displayed does not necessarily mean to be real– that we received from the MCS Evaluation team. As we can see, we have the results at the instance level for each agent (performer), this is what the measurement layouts is intended to predict. Let's provide some descriptions of the columns of our dataset:

- Task Name: indicates the name of the task that instance/scene belongs to.
- Cell: to understand this column, we have to take a look back at the concept of task hypercubes we defined in Section 3.1.1. This column identifies, given the task, in which cell from the hypercube the instance is assigned to. This summarises relevant information about the features of that instance. If we take the example of the occluded trajectory task instances that appear in Table 4.1, we notice that their cell value is D1. Then if we look at this task's hypercube, which can be seen in Figure 4.1, we can conclude that in that specific instance, the trajectory to get to the reward is straight and that the reward is located to the right with respect to the original position of the agent.
- Score: a binary variable indicating it the agent succeeded or not at the given instance –1 if it succeeded, 0 if not.
- Evaluation: indicates whether the instance comes from evaluation 6 or evaluation 7.
- Baseset: serves as an alias used by MCS for identifying instances.
- meta-feature *i*: these columns are the meta-features –binary variables– we used for characterising the tasks within the measurement layouts. These are: "moving object reasoning","core object reasoning", "quantify reasoning", "agent reasoning", "AI reorientation", "object permanence reasoning", "generalising", "tool use", "challenging navigation", "interactive task", "peripheral scene feature reasoning". Taking the definition of meta-features we provided back in the measurement layouts formalisation, these are cognitive demands and other high-level properties of the task that may assist on explaining agents' performance. When that meta-feature is present, the variable takes value 1, otherwise, 0.

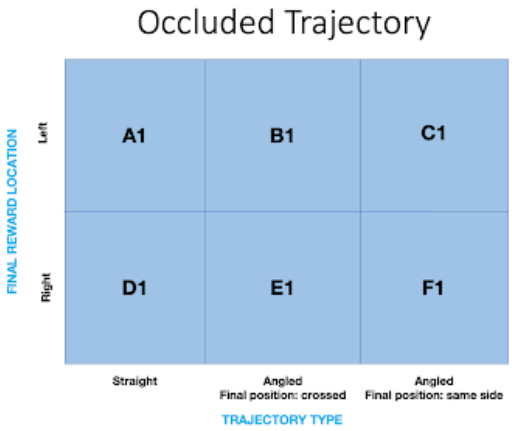

Figure 4.1: Occluded Trajectory Task Hypercube, taken from MCS programme website.

Performer	Task Name	Cell	Score	Evaluation	Baseset	meta-feature 1	...	meta-feature N
CORA	arithmetic	A1	1	Evaluation {6 | 7}	arithmetic_0010	1	...	0
MESS	arithmetic	A1	0	Evaluation {6 | 7}	arithmetic_0010	1	...	0
OPICS	arithmetic	A1	1	Evaluation {6 | 7}	arithmetic_0010	1	...	0
...	...	...	...	...	...	...	...	...
CORA	occluded trajectory	D1	0	Evaluation {6 | 7}	occluded_trajectory_0002	0	...	1
MESS	occluded trajectory	D1	0	Evaluation {6 | 7}	occluded_trajectory_0002	0	...	1
OPICS	occluded trajectory	D1	1	Evaluation {6 | 7}	occluded_trajectory_0002	0	...	1

Table 4.1: Evaluation Dataset Structure - Instance Level Data.

Performer	Task Name	Cell	Cell Mean	Evaluation	N Scenes	meta-feature 1	...	meta-feature N
CORA	arithmetic	A1	0.85	Evaluation {6 | 7}	25	1	...	0
MESS	arithmetic	A1	0.75	Evaluation {6 | 7}	25	1	...	0
OPICS	arithmetic	A1	0.9	Evaluation {6 | 7}	25	1	...	0
...	...	...	...	...	...	...	...	...
CORA	occluded trajectory	D1	0.65	Evaluation {6 | 7}	25	0	...	1
MESS	occluded trajectory	D1	0.45	Evaluation {6 | 7}	25	0	...	1
OPICS	occludded trajectory	D1	0.95	Evaluation {6 | 7}	25	0	...	1

Table 4.2: Evaluation Dataset Structure - Aggregated (Cell) Level Data.

On the other hand, Table 4.2 represents the same results but at the aggregated level. The aggregation is done grouping instances per cell value, task and agent. Then is computed the average performance –using the "Score" variable– per group. This percentage of success at the instances is referred to as "cell mean". The "N Scenes" column represents how many instances compose the group, i.e., how many scenes are used to compute the % of succesfully solved instances by the agent.

Now, let's present some exploratory analysis conducted to gain insights into how we could model performance based on the available meta-features with the measurement layouts. This analysis aimed to answer the following questions:

- Do any of the meta-features have predictive power for performance across different agents?
- Which agent is the best performer?
- Does any agent appear particularly strong or weak in specific instance settings?

4.1 Evaluation 6 Exploratory Analysis

In this case, we will start trying to find if there are specific meta-features from the scenes that seem to have predictive power about agent's performance. To do so, we used the Spearman's correlation coefficient to test whether there is a (monotonic) relationship between the performance at the aggregated level. Figure 4.3 correlation matrix has been obtained using aggregated level performance data coming from the three agents. Looking at the last column, we see that performance is poorly correlated with all the other variables, this results in meta-features having little predictive power about performance. It is also noticeable the strong negative association between Core Object Reasoning and Peripheral Scene Feature Reasoning; the positive correlations between Quantity Reasoning and Interactive Task; and between Challenging Navigation and Interactive Task. These associations provide an valuable insight of the patterns of demands we can find at the tasks from Evaluation 6. Moreover, it we put our focus on the last column again, specially to the cells associated to the the agents –in other words, the correlation between performance and a binary variable indicating that the response comes from that specific agent–, we see that at first glance the relative ordering of them in terms of being more or less predictable –absolute value of the correlation coefficient– would be CORA, OPICS and MESS.

When looking at these specific associations at the aggregated level per agent –Figures available in Appendix A.2–, we notice similar patterns to the ones observed aggregating all the agents. However, we could highlight that for CORA and OPICS agents, the Agent Reasoning demand has a significant negative impact on their performance –strong negative correlation–.

For Evaluation 6, the best performer is OPICS, followed by MESS and then by CORA, as Table 4.3 reveals.

Performer	Aggregated Performance
CORA	0.562
MESS	0.759
OPICS	0.893

Table 4.3: Aggregated Performance per Agents in Evaluation 6.

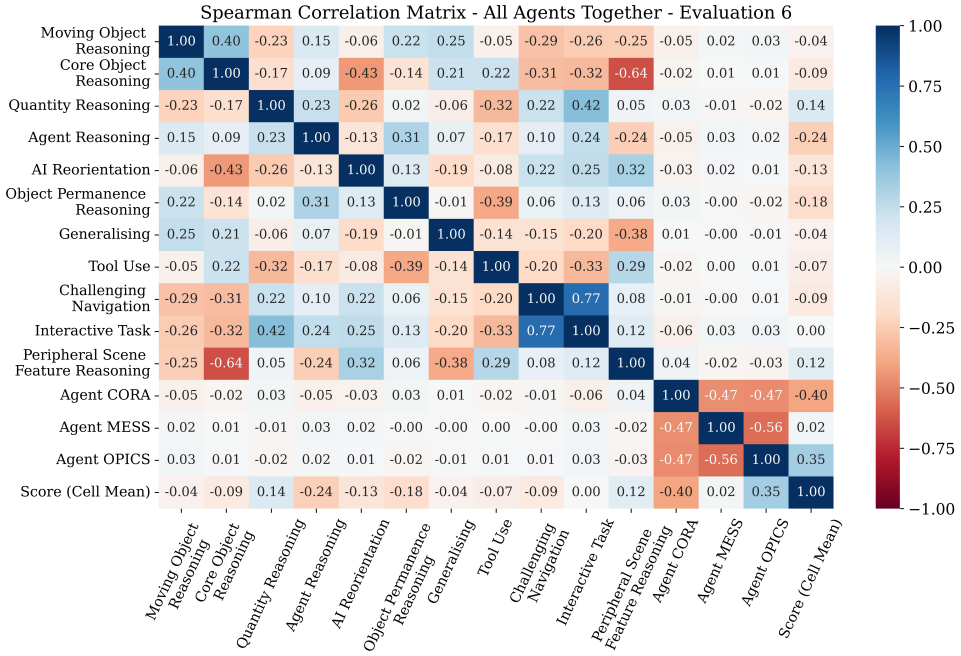

Figure 4.2: Spearman's Correlation Heatmap for All Agent Data (Aggregated Level) in Evaluation 6.

When analysing performance at the instance level, we used the Matthews Correlation Coefficient –Equation 4.1 provides the expression for this coefficient, its explanation and interpretation– to compute the association between cognitive demands and performance due to that both variable are binary. We can see the correlation matrix obtain in Figure A.2.

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{4.1}$$

where:

- TP = True Positives (C.Demand and Performance are both 1)
- TN = True Negatives (C.Demand and Performance are both 0)
- FP = False Positives (C.Demand is 0, Performance is 1)
- FN = False Negatives (C.Demand is 1, Performance is 0)

Notice that FP and FN could be defined inversely as they are expressed now, the interpretation of the coefficient would be the same. It is as follows:

- **MCC = 1**: A perfect positive correlation indicating that the presence cognitive demands consistently leads to high performance, and its absence demand consistently leads to low performance.
- **MCC = 0**: No correlation, indicating that cognitive demand has no association with performance.
- **MCC = -1**: A perfect positive correlation indicating that the presence of that cognitive demand consistently leads to bad performance, and its absence consistently leads to high performance.

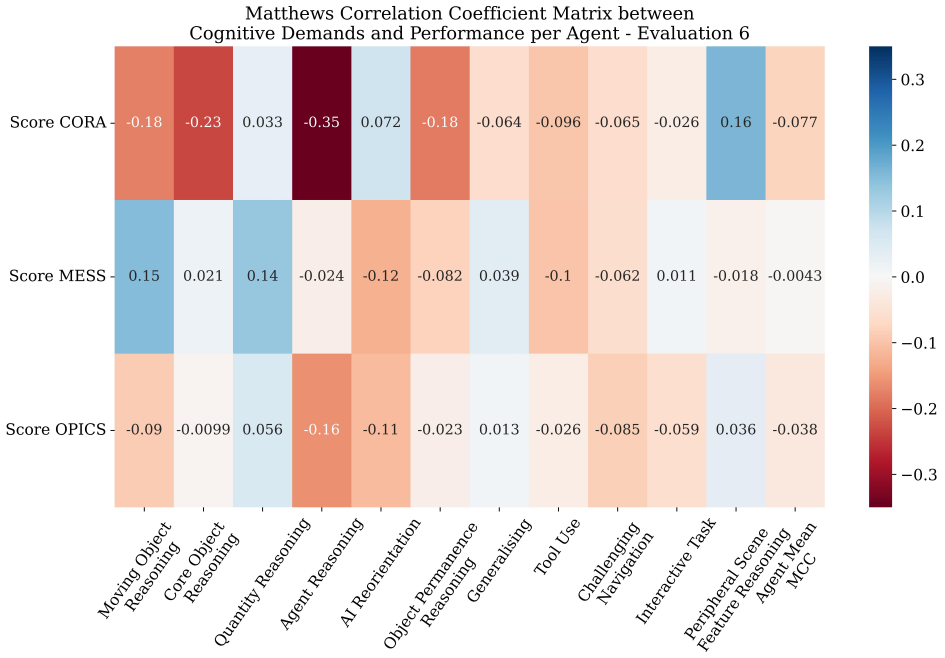

Figure 4.3: Matthews Correlation Coefficient between Demands and Performance (Instance Level) in Evaluation 6.

It can be remarked a similar appreciation we made when analysing data at the aggregated level: for CORA and OPICS –specially for CORA–, we see that the Agent Reasoning demand usually leads them to fail. However, this association is not very strong. Also, we see that Moving Object Reasoning demand influences negatively on CORA's performance, but conversely, when an instance has this type of demand, MESS agent usually thrives. Besides, we see that Core Object Reasoning affects negatively to CORA's performance.

None of the mentioned apparent associations are actually "significant", as the values are not very high in absolute value, but they might be providing us some intuition for understanding the capabilities that we will infer using the measurement layouts.

4.2 Evaluation 7 Exploratory Analysis

Similarly to what we observed with Evaluation 6 when analysing performance at the aggregated level, as Figure 4.4 depicts, again, there are not meta-features that have significant predictive power over agents' performance. Indeed, when we look at the cells corresponding to the agents, we see that predictability seems to be worse than in Evaluation 6. Now, the relative ordering in terms of predictability would be OPICS, CORA and MESS respectively. Moreover, we highlight again the negative association between the Core Object Reasoning and Peripheral Scene Feature Reasoning demands; and the positive association of the latter with Object Permanence Reasoning demands. We notice again that Challenging Navigation and Interactive Task meta-features appear together in instances often.

When looking at the agents' correlation matrices separately –available Figures in Appendix A.2–, it is remarkable to note that differently to Evaluation 6, Agent Reasoning does not seem to influence importantly to any of the agents' performance anymore. Nonetheless, we notice that OPICS performance seems to be negatively correlated with the Moving Object Reasoning and Tool Use demands.

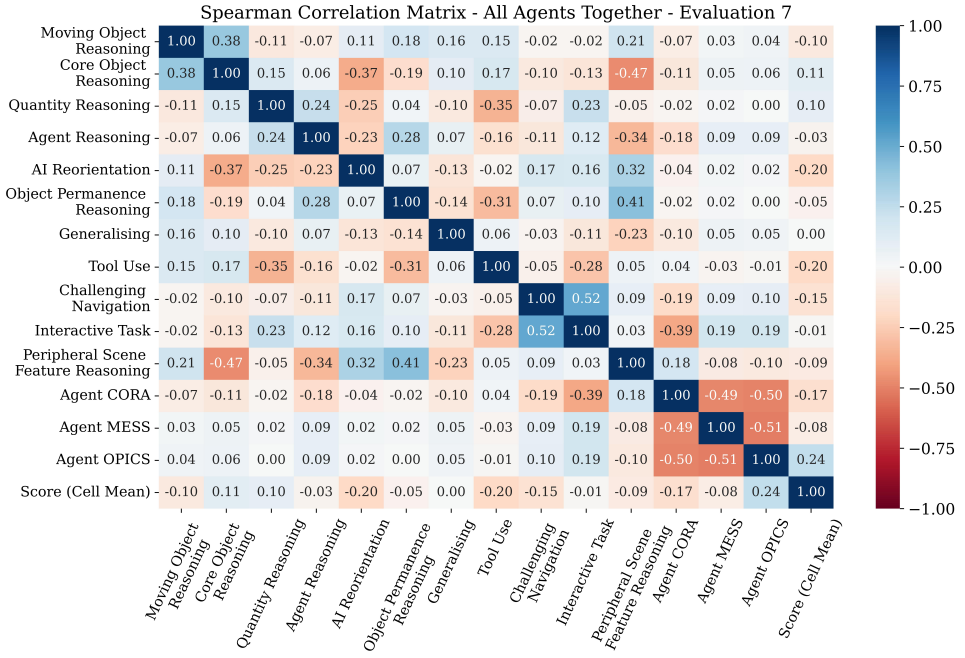

Figure 4.4: Spearman's Correlation Heatmap for All Agent Data in Evaluation 7.

For Evaluation 7, the best performer is again OPICS, but in this case, CORA performs better than MESS, as Table 4.4 shows.

Performer	Aggregated Performance
CORA	0.707
MESS	0.691
OPICS	0.790

Table 4.4: Aggregated Performance per Agents in Evaluation 7.

When analysing the Matthews Correlation Coefficient at the instance level between performance and demands –Figure 4.5, we confirm the observations made at the aggregated level, i.e. it is observed negative correlation coefficients between Tool Use and Moving Object Reasoning demands with respect to performance in the case of OPICS agent. Nevertheless, if we look at the last column, which represent the average correlation across demands with respect to performance, it reveals a lack of predictability –this is probably due to the fact that correlations fluctuate and some are positive while others are negative, what eventually leads to an average MCC close to 0 for the three agents.

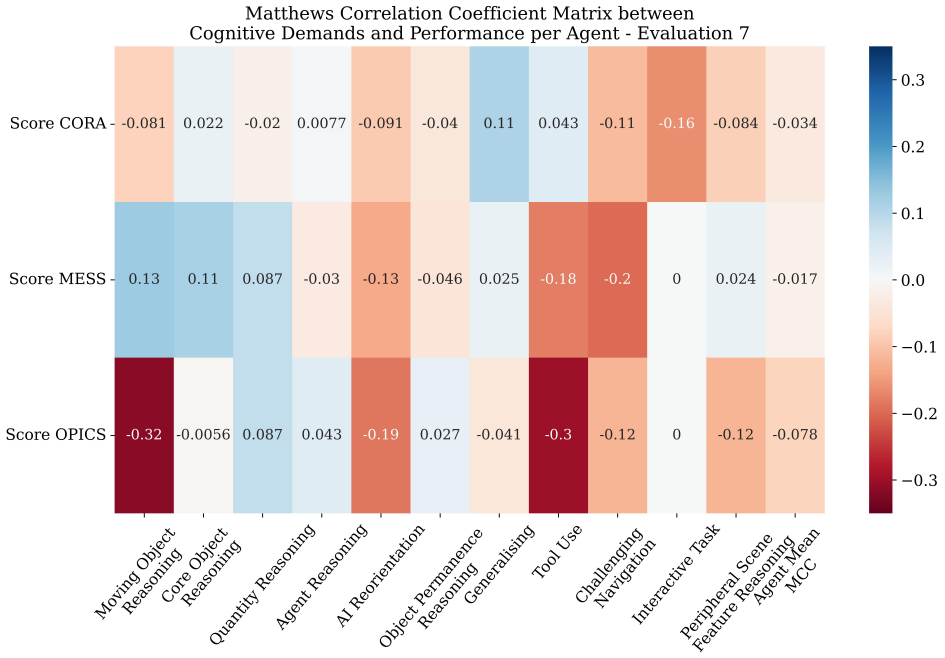

Figure 4.5: Matthews Correlation Coefficient between Demands and Performance (Instance Level) in Evaluation 7.

However, in general terms, the observed correlations both at the aggregated and instance level do not seem enough "strong" to extrapolate the conclusions and make statements about systems capabilities.

4.3 Exploring Agents' "Weaknesses" and "Strengths"

In this case, we decided to explore the idea of studying the distribution of performance at the aggregated level and by grouping by task in order to build some intuition about the possible capabilities of the agents. Also, we differentiate per Evaluation "act" to study how the cognitive profiles that we would build through the measurement layouts could fluctuate their predictions on the estimated capabilities from one evaluation to another.

We pretended to detect "strengths" or "weaknesses" by studying distributions of aggregated performance per task type. If we observed that an agent had a consistent outstanding performance distribution in a given task at both evaluations, we considered that the agent was likely to have advanced capabilities related to the cognitive demands that were present in that task. For instance, if we observed that agent CORA achieved close to a perfect score for a task –i.e. the distribution of the aggregated performances per cell for that given task is very narrow and close to 1–, we would suspect that CORA has advanced capabilities related to the demands from this task. For detecting weaknesses, we proceeded exactly in the same way, but looking for narrow distributions around relatively low average performance.

For studying these distributions, we took aggregated level results –that if we remember, are a result of averaging performance after grouping instances by cell and task, so, for each task we have an observation per cell, which represents the average performance at instances located at that cell in that task–. Then, we grouped results by task, and we represented the distribution as boxplots of them at both evaluations. We will only show those tasks from which we got valuable insights. We found the following results:

Figure 4.6 shows the distribution of cell aggregated performance at Interactive Object Permanence task per Agent at the different Evaluations. Here we highlight that both MESS and OPICS excel at most of the instances from this task at both evaluation acts. However, in the case of CORA, we highlight that there is a notable improvement from Evaluation 6 to Evaluation 7, but average performance is considerably poor, specially comparing it to the other two agents.

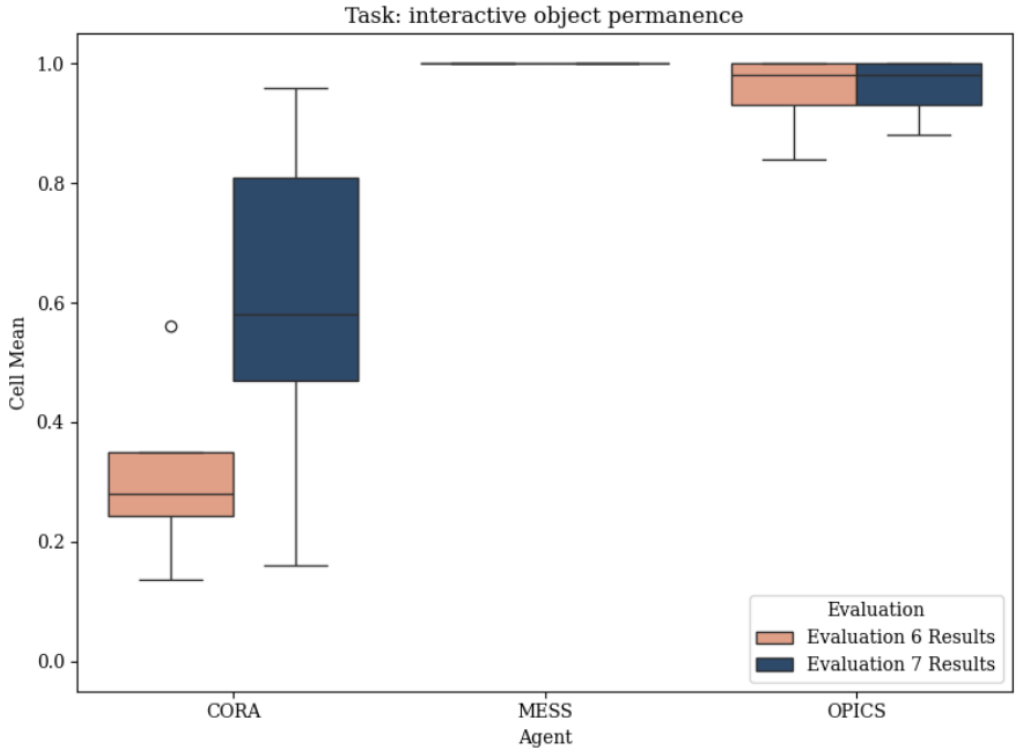

Figure 4.6: Distribution of Cell Aggregated Performance at Interactive Object Permanence Task.

For each meta-feature we had, we computed the percentage of instances from this task in which that property/cognitive demand was present. This is summarised in Table 4.5.

Task	% of Instances
Moving Object Reasoning	100
Core Object Reasoning	100
Quantity Reasoning	0.00
Agent Reasoning	0.00
AI Reorientation	0.00
Object Permanence Reasoning	48.45
Generalising	0.00
Tool Use	0.00
Challenging Navigation	100
Interactive Task	100
Peripheral Scene Feature Reasoning	0.00

Table 4.5: Percentage of "presence" of meta-features in Instances from Interactive Object Permanence Task.

Given that the cognitive demands of Moving and Core Object Reasoning, as well as the high-level properties of Challenging Navigation and Interactive Task, are present in all instances of this task, and both MESS and OPICS excel at it, we could infer that when

modeling capabilities through the measurement layout, the ones assigned to tackling these demands might be expected to be advanced for them, at least in comparison to the CORA agent.

We also see a very similar observation for agents MESS and OPICS in the case of the task "Moving Target Prediction Task", they are consistently very good performers at instances from this task at both Evaluations, as it can be seen in Figure 4.7. Note that CORA, was not tested on this task at Evaluation 6.

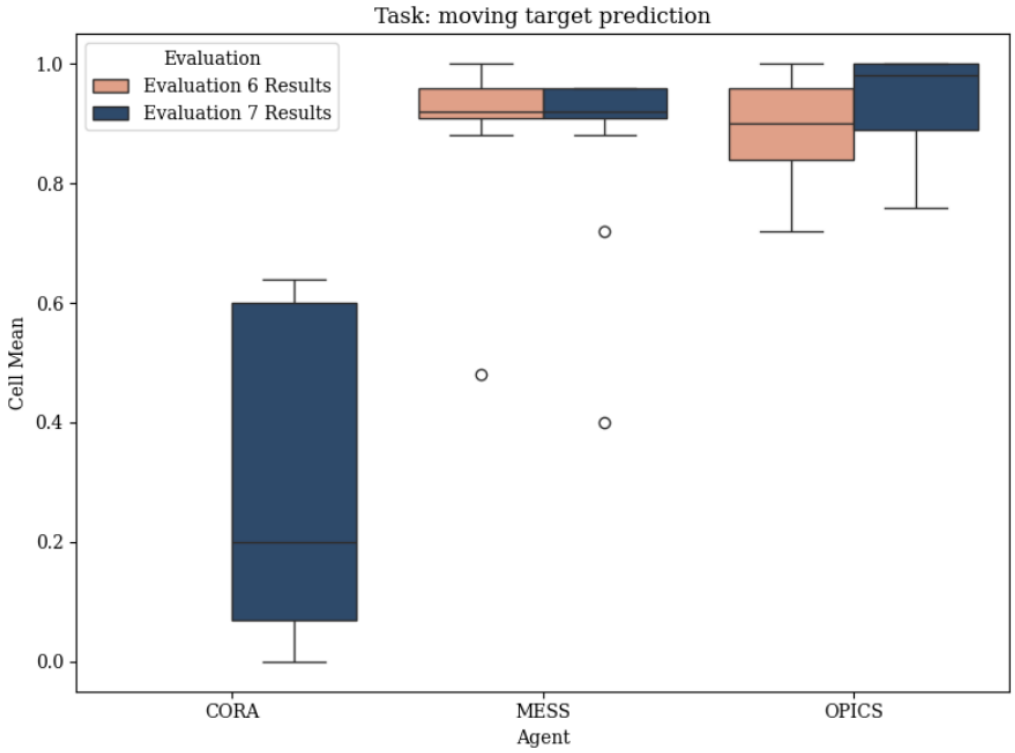

Figure 4.7: Distribution of Cell Aggregated Performance at Moving Target Prediction Task.

Therefore, observing in Table 4.6 the relative frequency with which demands are distributed through the instances from this task, we could lead us to "make a guess" on MESS and OPICS possessing advanced cognitive abilities that address the cognitive demands of Moving Object Reasoning, Core Object Reasoning, AI Reorientation, and Interactive Task. In the case of agent CORA, we cannot draw any conclusions from this task, as the aggregated performance per cell has a broad distribution.

For the Spatial Elimination task, we observe in Figure 4.8 that all three agents are strong performers, with the mean of their cell aggregated performance distributions at both evaluations being around 100% accuracy/success.

After having observed that all three agents excel at this task, it is reasonable to consider the capabilities modelled later in the measurement layouts at both evaluations to address the demands present in 100% of the instances from this task –such as Core Object Reasoning, Quantity Reasoning, Challenging Navigation, and Interactive Task– as advanced.

When considering the "Obstacle" task, we see again that the three agents are very good performers –note that neither MESS agent was not evaluated on this task at the Evaluation 7, nor OPICS at Evaluation 6–. This can be seen by looking at their aggregate performance distributions in Figure 4.9, because they have very narrow distributions with considerably high average performance.

Task	% of Instances
Moving Object Reasoning	100
Core Object Reasoning	100
Quantity Reasoning	0.00
Agent Reasoning	0.00
AI Reorientation	100
Object Permanence Reasoning	0.00
Generalising	0.00
Tool Use	0.00
Challenging Navigation	50.0
Interactive Task	100
Peripheral Scene Feature Reasoning	0.00

Table 4.6: Percentage of "presence" of meta-features in Instances from Moving Target Prediction Task.

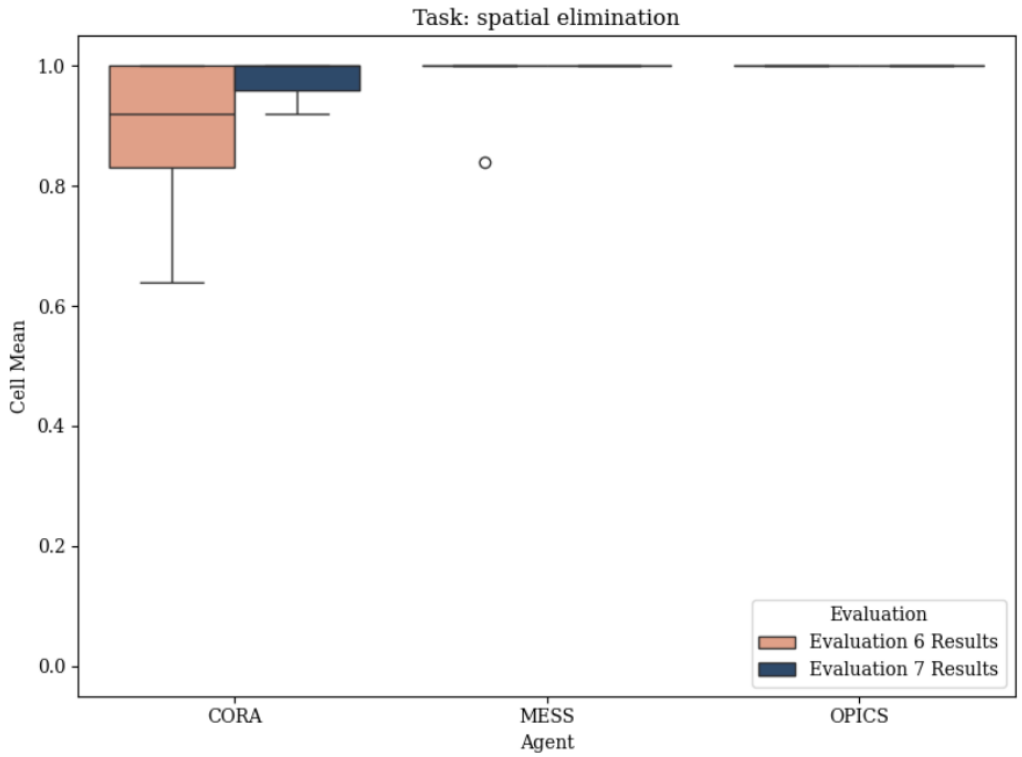

Figure 4.8: Distribution of Cell Aggregated Performance at Spatial Elimination Task.

Task	% of Instances
Moving Object Reasoning	0.00
Core Object Reasoning	100
Quantity Reasoning	100
Agent Reasoning	0.00
AI Reorientation	0.00
Object Permanence Reasoning	25.0
Generalising	0.00
Tool Use	0.00
Challenging Navigation	100.0
Interactive Task	100
Peripheral Scene Feature Reasoning	0.00

Table 4.7: Percentage of "presence" of meta-features in Instances from Spatial Elimination Task.

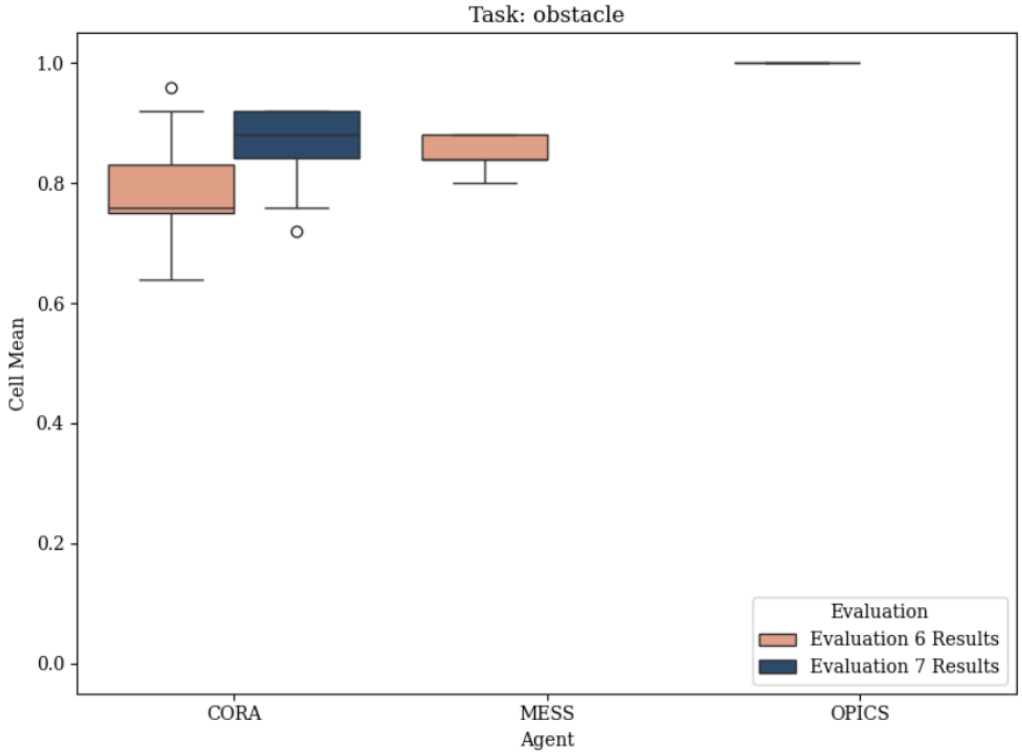

Figure 4.9: Distribution of Cell Aggregated Performance at Obstacle Task.

Task	% of Instances
Moving Object Reasoning	0.00
Core Object Reasoning	100
Quantity Reasoning	0.00
Agent Reasoning	0.00
AI Reorientation	0.00
Object Permanence Reasoning	50.0
Generalising	50.0
Tool Use	0.00
Challenging Navigation	50.0
Interactive Task	100
Peripheral Scene Feature Reasoning	0.00

Table 4.8: Percentage of "presence" of meta-features in Instances from Obstacle Task.

Then, by looking at the presence of the demands in this task instances in Table 4.8, we could suspect that these agents could have robust capabilities for addressing instances demanding in Core Object Reasoning and that are categorised as a Interactive Task.

In this last task that we will consider we put the focus on agent CORA, as we cannot reliable make some assumptions about agents MESS and OPICS because despite their performance distributions are narrow and on average close to 100% accuracy, there are some cells for them where the observed performance is completely the opposite, very erratic and they practically fail on every instance from that cell –this can be seen as the "isolated" points represented as outliers in their boxplots.

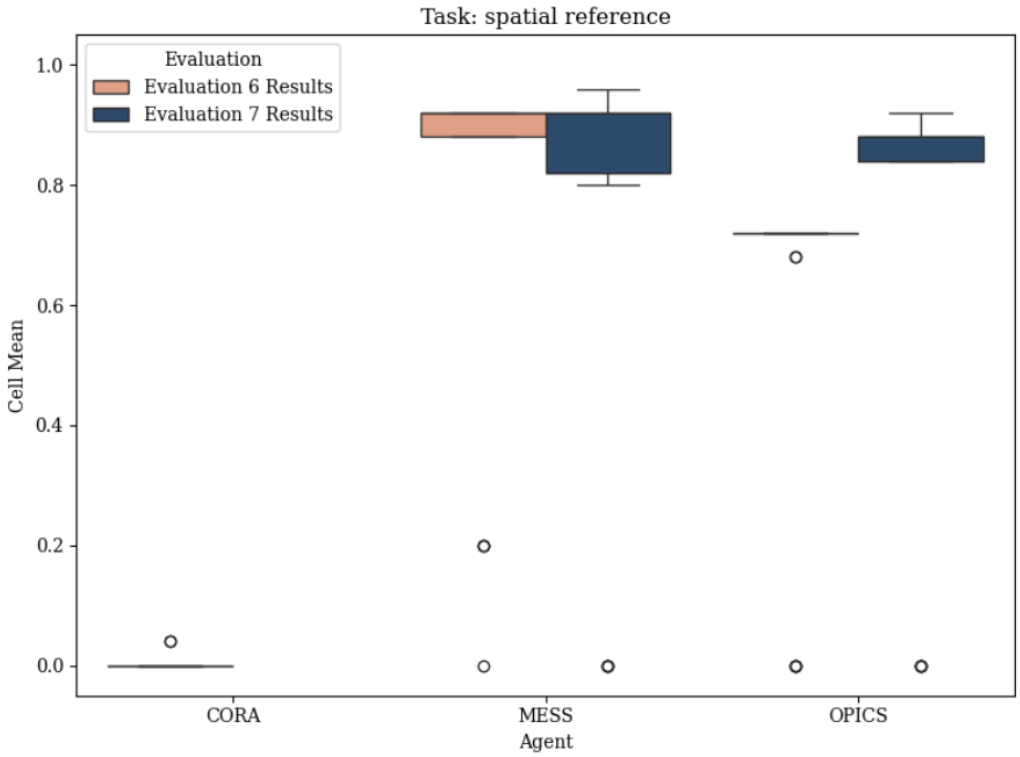

Figure 4.10: Distribution of Cell Aggregated Performance at Spatial Reference Task.

Task	% of Instances
Moving Object Reasoning	100
Core Object Reasoning	100
Quantity Reasoning	55.5
Agent Reasoning	100
AI Reorientation	0.00
Object Permanence Reasoning	100
Generalising	40.7
Tool Use	0.00
Challenging Navigation	100
Interactive Task	100
Peripheral Scene Feature Reasoning	0.00

Table 4.9: Percentage of "presence" of meta-features in Instances from Spatial Reference Task.

It is important to note that CORA was not assessed on this task in Evaluation 6. Therefore, the conclusions drawn are only applicable to CORA's performance specifically in Evaluation 6. Since CORA failed in nearly every instance of this task, and considering

the frequency of demands across its instances as indicated in Table 4.9, we could speculate that CORA likely has limited capabilities related to Agent Reasoning, Moving and Core Object Reasoning, Object Permanence Reasoning, Challenging Navigation, and Interactive Task. This is due to that these demands were consistently present in all scenes of this task.

In this final section of the exploratory analysis, we have speculated that individuals who excel in certain tasks may possess advanced capabilities specifically suited to address the cognitive demands that appear more consistently in that task. Conversely, poor performers may struggle with these demands and, therefore have limited capabilities to "counter" this demands. This "speculative" approach may sound very familiar indeed, and this is because it is close to the intuition behind the bottom-up inference we introduced with an example in Section 3.3 and that is carried out by the measurement layouts.

Here, our shallow assumptions –which might be wrong– about agents capabilities were based exclusively on some specific tasks, due to the uncertainty and lack of a clear perspective when considering broader performance and demand distributions of other tasks. Differently to this speculative approach, measurement layouts excel in taking results across all instances and tasks but with the possibility of analysing one agent at-a-time. The comprehensive method we introduce in this project is able to consider all the nuances of an agent's performance across different task instances –what includes their demands–, providing a precise inference of their capabilities.

CHAPTER 5
Experimental Setting

In this chapter, it is introduced how we conducted the experiments with the measurement layouts; how we tested and compared its predictive performance against an "assessor" and the baseline prediction; and which kind of settings of this framework we considered for inferring MCS programme agents capability profiles.

5.1 Measurement Layouts for Inferring Capability Profiles and Predicting Performance

As we explained thoroughly in Section 3.1.1, the two evaluation acts that we are considering generated meta-annotated instance level results for the three agents. Some tasks overlapped between evaluations, but not all the agents had exactly the same scenes to be evaluated in. We will use instance level and aggregated results per cell –see Tables 4.2 and 4.1 for an illustration of their structure– to infer the cognitive profile of the agents CORA, MESS and OPICS and predict their future performance.

We will show the results of two different settings of the measurement layouts per evaluation data, which are fitted for each combination of agent and data granularity level –instance or aggregated level. Depending on the target response, i.e. whether the data is taken at the instance level or aggregated level, the measurement layouts is trained to predict the success or fail of an agent at an instance, or a continuous response (from 0 to 1) at the aggregated level, which represents the average success at a given cell of the corresponding agent.

After an iteration of fitting a measurement layouts to an agent's data –it does not matter whether it is aggregated or instance level data– with HMC algorithm, we obtain the following:

- The capability profile of the agent. Actually, what it is obtained is the estimation from the HMC algorithm for the mean of the approximated marginal distribution of the abilities that we set the cognitive profile to be composed by.

- The performance on held-out data given by a selected metric we will introduce in the next section.

More details into the fitting and evaluation process of the measurement layouts in the following section.

5.2 Fitting and Evaluating Measurement Layouts Predictive Performance

To fit a measurement layouts it is followed this process:

1. Define the model. As a reminder measurement layouts are specialized HBNs that incorporate tools like linking and compensatory functions to model the hierarchical dependencies between agents' capabilities and demands.
    - First, select the priors for the capabilities that compose the cognitive profile to be inferred.
    - Define the meta-features from instances that will be introduced in the measurement layout.
    - Specify the inner details of the topology for the measurement layout, such as which margins are considered; whether these margins generate "sub-task" performances; determining whether capabilities are dependent or independent and selecting which linking functions to use for combining dependent capabilities.
    - Choose the type of noise that will be introduced.
    - Depending on the granularity of data (instance or aggregated level), define the target variable as a Bernoulli or a Beta distribution, with parameters depending on the hierarchical dependencies of capabilities and demands encoded "up" in the measurement layout.
2. Select the number of iterations of the Hamiltonian Monte Carlo (HMC) algorithm that will be executed and the hyperparameters of the algorithm.
3. For each iteration, split the agent's data into training and test sets –using a 90/10 split for training and testing respectively. Just as a reminder, the measurement layouts is fitted for each agent separately.
4. In each iteration, HMC uses training data to approximate our parameter distributions. After fitting them –including the estimations for the capability profile–, it uses the posterior predictive to compute the predictions on the test split.
5. For evaluating the predictive performance of the measurement layout, we use the Brier Score. See Subsection 5.2.1 for the explanation of this choice.

For testing a given setting of the measurement layouts in an agents' evaluation performance data we predetermine the following considerations for the HMC algorithm:

- Five iterations of the HMC algorithm will be executed.
- Each iteration of the algorithm will consist of 5 chains. For each chain, there will be 1000 samples in the burn-in period and 2000 samples used to approximate the parameter distributions.
- For the posterior predictive distribution, we will also use 5 chains of 2000 samples each. The final prediction for each item is the average sampled value for it across the chains.
- In the case of instance-level data, despite the measurement layouts using a Bernoulli variable as the output, the final output can be regarded as the estimated probability of success for a given instance.

- In the case of aggregated data, the prediction for a cell is the estimated average performance at the cell.

5.2.1. Brier Score

The Brier Score is defined as:

$$\text{Brier Score} = \frac{1}{N}\sum_{i=1}^{N}(p_i - o_i)^2 \tag{5.1}$$

We use Brier Score because of its probabilistic interpretation. In the case of the measurement layouts being used for estimating the probabilities of an agent succeeding in an instance, it reflects how close the estimated probabilities are to the real observation (0 or 1). Similarly, when the model is predicting the average performance of an agent at a given cell, this is analogous to predicting probabilities for individual instances, so the Brier Score is a good metric to express how accurate the measurement layouts estimations are.

The Brier Score can be decomposed into calibration and refinement components:

$$\text{Brier Score} = \underbrace{\frac{1}{N}\sum_{j=1}^{K} n_j(\hat{p}_j - \hat{o}_j)^2}_{\text{Calibration}} + \underbrace{\frac{1}{N}\sum_{j=1}^{K} n_j\hat{o}_j(1-\hat{o}_j)}_{\text{Refinement}} \tag{5.2}$$

where:

- N is the total number of instances (depends on the agent and whether the data is at the instance level or aggregated per cell).
- K is the number of bins (in our case 10).
- n_j is the number of predictions in bin j.
- $\hat{p}_j$ is the average predicted probability (or proportion in the aggregated case) in bin j.
- $\hat{o}_j$ average observation (represents a probability in the instance level case, and a proportion in the aggregated level) in bin j .

The reason for this decomposition is because the components provide a complementary comprehensive interpretation of the Brier Score. The calibration component provides a measure of how close the forecast probabilities –or proportions– are close to the true ones. Meanwhile, the refinement component reflects the measurement layout's ability to differentiate between instances with varying probabilities of success. A higher refinement value indicates that the outcomes are more evenly split (closer to 0.5), while a lower value indicates more certainty (closer to 0 or 1). Basically, the latter measures the inherent uncertainty/variability in a given bin. [4]

5.3 Predictive Performance Comparison

To provide a comparison of predictive performance to the measurement layout, we chose a baseline prediction and an assessor:

- Baseline Prediction: we take the approach of extrapolating the success ratio of each agent as the prediction.
  - Instance Level Baseline Prediction: for each iteration of the algorithm, it is taken agent's training data to estimate the average performance. This mean performance metric is used as the prediction for the held-out instances of that iteration.
  - Aggregated Level Baseline Prediction: Exactly the same, but in this case the average performance represents the mean of aggregated performance per cell task.

    At the end the interpretation is the same, is used the average performance as extrapolation of future performance.
- XGBoost Assesor: XGBoost or "Extreme Gradient Boosting" is one of the most widely used Machine Learning algorithms. It leverages gradient boosted trees and is renowned for its efficiency and predictive power, especially with tabular data. XGBoost builds an ensemble of decision trees following the technique of gradient boosting, where new models are added iteratively to correct the error of previous existing models [17]. The model takes as input the same meta-features as the measurement layouts does. Depending on the granularity of the data we use, the objective function for fitting the XGBoost model varies:
  - Instance Level Data: it has a binary logistic objective, as it is used for a binary classification task –the model predicts the probability of success (e.g., an agent's success) given the meta-features of the instance.
  - Aggregated Level Data: it is a regression problem. Here, the objective is to predict the aggregated performance per cell.

For comparing them to the measurement layouts we also use the Brier Score as the evaluation metric.

5.4 The Measurement Layouts General Topology for MCS

As explained in Section 3.3, "building" the measurement layouts topology involves selecting the abilities for the cognitive profile based on available meta-features, structuring hierarchical dependencies, and determining whether the capabilities are compensatory. This process heavily relies on domain knowledge in cognitive science and development psychology, limiting our flexibility to modify the framework. Furthermore, our ability to vary the architecture is constrained by the nature of the tasks used to assess the agents' cognitive profiles. Consequently, the MCS Evaluation team and the RECOG-AI team, with whom we collaborated on this project, predefined the cognitive profile and its dependencies on the available demands and task properties. This topology can be seen visually in Figure 5.1.

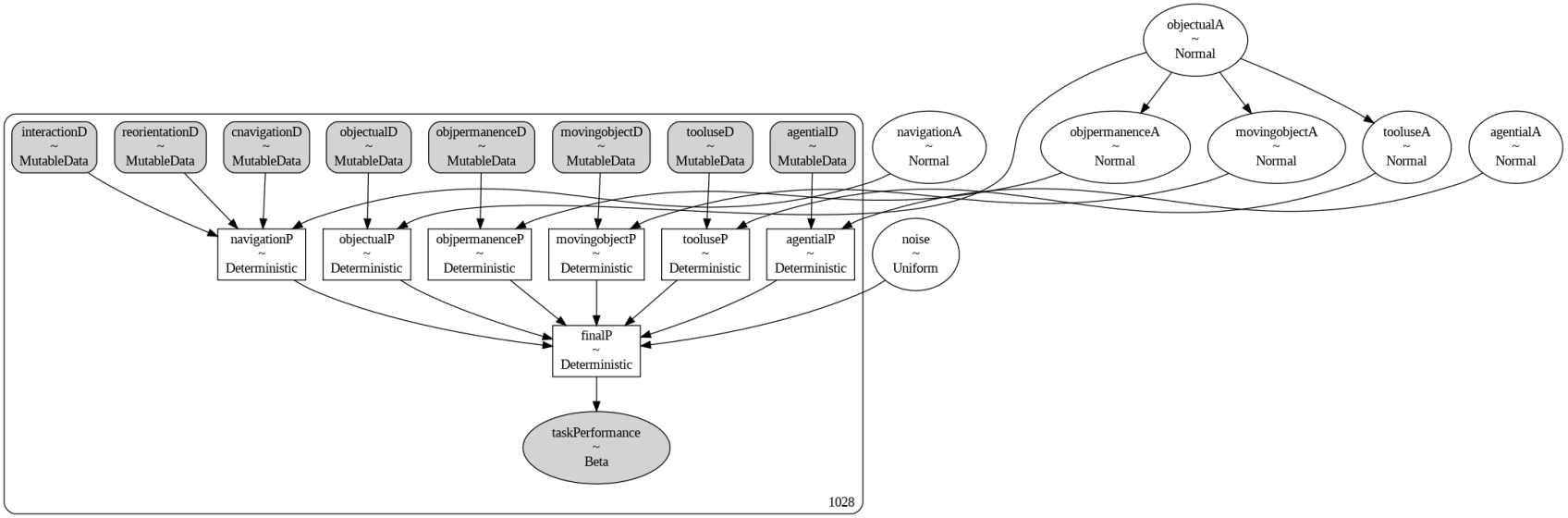

Figure 5.1: Initial Measurement Layouts Topology.

We notice that the meta-features appear as root (rectangular shaded) nodes of the HBN, which correspond to the annotated macro-level variables by the MCS Programme Evaluation team. Also, the cognitive profile nodes (white circular) are also the roots of the HBN. The capability profile is composed by the Navigation Ability –represented as navigationA–, the Object Permanence Ability –represented as objPermanenceA–, Moving Object Ability –represented as movingobjectA–, the Tool Use Ability –represented as tooluseA–, the Agential Ability –represented as agentialA– and the Objectual Ability –represented as objectualA–. Note that Object Permanence, Moving Object and Tool Use abilities distributions will be dependent on Objectual Ability distribution. More specifically, the mean of their respective distributions depends on the distribution of the Objectual Ability.

The abilities appear to fall into three distinct categories, each corresponding to a specific domain studied in the MCS Programme. Navigation Ability is linked to the Places Domain and is associated with Interaction, Reorientation, and Complex Navigation Demands. Objectual Ability focuses on the Objects Domain, addressing Objectual, Object Permanence, Moving Object, and Tool Use demands. Finally, Agential Ability targets the Agential Demand, and, as suggested by its name it pertains to the Agents Domain.

The rest of the topology characteristics is a specification, to introduce the "core" of our measurement layouts structure we just needed to introduce the predefined cognitive profile, the metafeatures and the dependencies between them from a high-level perspective.

CHAPTER 6
Results

In this chapter, we analyse the results obtained during the experimental phase using measurement layouts for inferring MCS Programme agents' capability profiles and predicting their performance. To begin with, we will start introducing the two measurement layouts topologies we will consider. The results are presented following the sequence of evaluation acts, starting with Evaluation 6 and then proceeding to Evaluation 7. For each evaluation, we will examine the two presented variations of the measurement layouts. We will analyse their predictive performance at both the instance and aggregated levels, interpret the inferred cognitive profiles of the agents by looking at their radial plots, and explore how agents are ranked based on their capabilities relative to their observed performance. Additionally, we will conduct a comparative analysis of the two measurement layouts topologies considered. Notice that for each radial plot we will provide the actual inferred mean of the distribution for the capabilities in a table, together with their estimation error –standard deviation–. However, for making it more straight and easier to link the approximated capabilities to the predictive performance of the measurement layouts setting we only include the radial plot in this section. For inspecting the tables refer to Appendix C.

After studying each evaluation separately, we will analyse the possible fluctuation/-consistency of the inferred capability profiles across evaluations, the measurement layouts predictive power and its dependency to the granularity of data used.

6.1 The Measurement Layouts Settings

In Section 5.4 we introduced the fundamental part of the measurement layouts topology we will use for inferring the capability profiles of the MCS Programme Agents. These were the aspects of the measurement layouts that would remain unaltered for the rest of the experimentation process –the elements of the cognitive profile, the meta-features of the tasks and the basic dependencies between them. Now, we will present the details of the two settings we consider for the assessing its predictive power and analysing the inferred capability profiles.

6.1.1. Setting 1: Normal Priors and "Downscaling" Noise

The reasoning behind choosing normal priors for the distribution of the capabilities is the following:

- Defines an unconstrained scale for the capabilities, allowing to express the absence of an ability through the approximation of the capability as a negative value.

- Following the measurement theory explained in Section 3.3, the choice of this prior defines an interval scale for the capabilities. This implies that the scale gives the same probability of success when when the difference between capability and demand is the same –e.g., capability 10 and demand 8 versus capability 5 and demand 3.

- We can use the capabilities inferred to estimate the extent to which an agent is more "capable" than another.

With respect to the choice of noise, we use a variant of the noise introduced by Equation 3.24 in Section 3.3. In this specific noise setting, we delete the term $\varphi_j \nu_j$. The interpretation of the inferred values for the parameters of the distribution of noise –φ_j in the first term of the equation– is that it is "capping" the internal performance. For example, a value of 0.2 inferred for the noise would mean that the final estimated performance could be 0.8 as maximum (1-0.2).

For computing non-observed intermediate performances –navigationP, objectualP, objPermanenceP, movingobjectP, tooluseP and agentialP nodes in 5.1– through the logistic function of margins, we use the margin for binary demands –given that our demands are binary– expressed by Equation 3.22 in Section 3.3 and its respective logistic function expressed by Equation 3.23.

Due to that we do not consider a compensatory setting, these intermediate performances are combined through their product. These product is then fed to model final performance distribution parameter, which is also affected by the noise variant we explained above.

6.1.2. Setting 2: Scaled Beta Priors and "Convex Combination" Noise

Selecting a scaled beta prior for the capabilities distribution has the following consequences on the measurement layouts inferences:

- Capabilities are now bounded between 0 and 2.

- Following the measurement theory explained in Section 3.3, the capabilities are now in an ordinal scale.

- Since the capabilities are on an ordinal scale, the final inferred mean of their beta distribution cannot be used to quantify the magnitude of differences between agents' capabilities. However, it can provide an idea of the relative ordering of general capability based on the inferred distributions for them.

The selection of noise is represented by the same Equation as the previous setting, but in this case, we preserve the $\varphi_j \nu_j$ term. However, ν_j is not a "learnable" parameter through the HMC algorithm, but it is fixed to the average observed performance of that agent. This means that the inferred mean of the distribution for noise (φ_j) determines to which extent the estimation of future performance relies on observed performance. For instance, if hypothetically the approximated noise is 1, all the future predictions of the measurement layouts will depend on the observed responses from that agent.

For this setting we consider that capabilities are non-compensatory as well.

6.2 Evaluation 6: Measurement Layouts Setting 1

As shown in Figure 6.1, it is notable that despite MESS having advanced Moving Object and Agential capabilities compared to OPICS, the best performer, MESS shows worse performance from both the instance and aggregated level perspectives. This is evidenced by the Performance column in Tables 6.1 and 6.2. Conversely, OPICS is estimated to have superior Object Permanence and Objectual capabilities. Interestingly, MESS has similar estimated values for these capabilities to CORA, the worst performer. This disparity suggests that the evaluation instances in Evaluation 6 may not heavily feature items demanding Agential and Moving Object abilities, but rather items requiring Objectual Ability are more prevalent.

For CORA, the lack of Agential capability is evident, as its estimated mean for this ability distribution is negative. Additionally, it is interesting to note that both CORA and MESS have Tool Use capabilities close to zero. In contrast, OPICS has an estimated Tool Use capability nearly double that of the other two systems, which may also explain its better performance. On the other hand, it seems that the three agents have very similar Navigation capabilities.

In summary, the Evaluation 6 appears to favour items requiring Objectual and Tool Use abilities, which align more closely with OPICS's strengths, what may explain its superior performance in comparison to CORA and MESS.

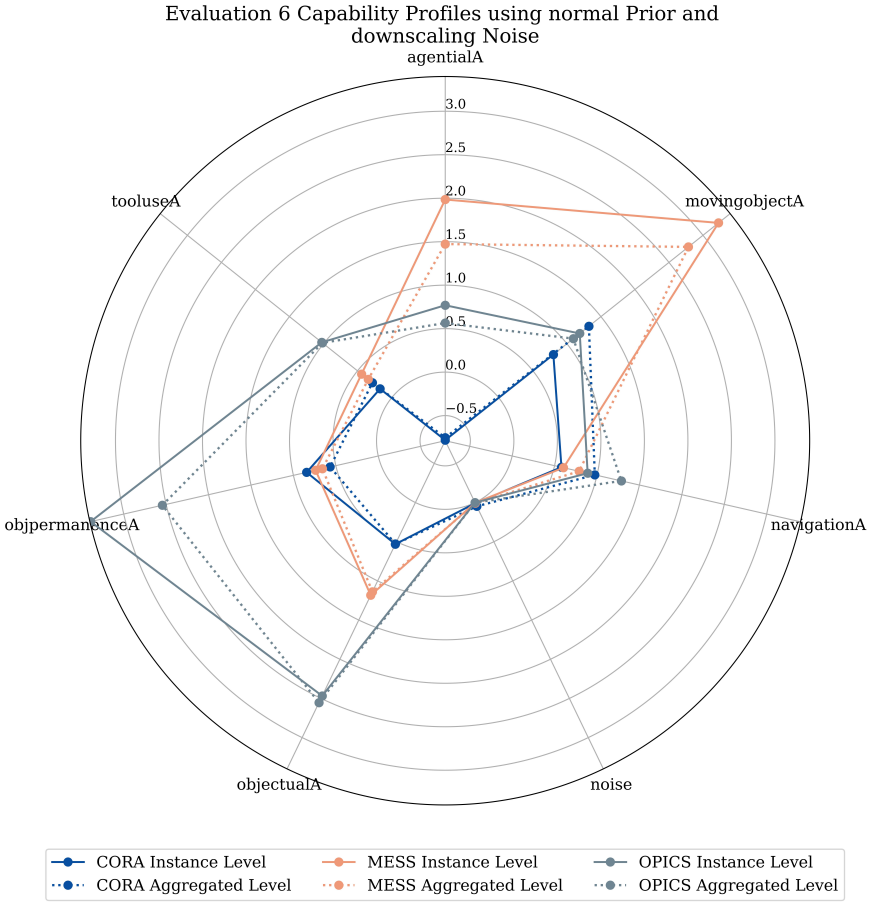

Figure 6.1: Radial Plot Capability Profiles from Evaluation 6 with Measurement Layouts Setting 1.

With respect to the predictive performance of the Measurement Layout, we can refer to Tables 6.1 and 6.2, which present the data at both the instance and aggregated levels respectively. We observe that the measurement layouts consistently outperforms the estimations based on observed aggregate performance at both levels. While its estimations are slightly worse than those provided by the assessor at the aggregated level, the Measurement Layouts surpasses the predictive performance of XGBoost at the instance level.

Another strength of this setting we are considering of measurement layouts is that it provides robust noise estimations, consistently close to zero across both granularities of evaluation data –instance and aggregated. This promising finding suggests that, with a sufficient and varied performance dataset, the predefined cognitive profile we are assuming within the measurement layouts can effectively explain the agents behaviour.

Agent	Performance	Aggregate (BS)	Measurement Layouts (BS)	Assessor (BS)
CORA	0.5716	0.2450	**0.1991**	0.2823
MESS	0.7589	0.1829	**0.1729**	0.2153
OPICS	0.8987	0.0885	**0.0846**	0.0970
Mean	0.7385	0.1721	**0.1522**	0.1980

Table 6.1: Predictive Performance (Brier Score) from Measurement Layouts Setting 1 on Evaluation 6 - Instance Level.

Agent	Performance	Aggregate (BS)	Measurement Layouts (BS)	Assessor (BS)
CORA	0.5716	0.0671	0.0641	**0.0609**
MESS	0.7589	0.0933	0.0773	**0.0522**
OPICS	0.8987	0.066	0.0296	**0.0224**
Mean	0.7385	0.0837	0.0570	**0.0440**

Table 6.2: Predictive Performance (Brier Score) from Measurement Layouts Setting 1 on Evaluation 6 - Aggregated Level.

6.3 Evaluation 6: Measurement Layouts Setting 2

Figure 6.2 provides insights into the capability profiles of the agents using scaled beta priors. With capabilities bounded between 0 and 2, and simulating an ordinal scale, we can infer the relative ordering of the agents' general capabilities. We highlight MESS shows strong Agential and Moving Object capabilities, similarly to what we observed with the previous setting. However, its overall performance, once again indicates that Evaluation 6 instances may not emphasise tasks requiring these abilities. On the other hand, OPICS, with more advanced Object Permanence and Objectual capabilities, exhibits higher performance. This appreciation reinforces the intuition we observed analysing the first setting on this evaluation data, that the tasks in which agents were assessed on this evaluation act, featured more instances demanding from the point of view of Objects domain –area where OPICS seems to excel compared to the others to MESS and CORA– than from other common sense domains considered.

Interpreting the capability profiles inferred assuming Beta priors is complex, and we cannot make accurate assumptions about how much more capable OPICS is –or at least seems– compared to MESS and CORA. This complexity comes because the Beta prior does not reveal significant differences on the capability profile, and we can only rely on the relative order of the approximations computed for the capabilities. In this sense, we

can confirm that the relative ordering of agents' capabilities observed in the first setting appears to be consistent.

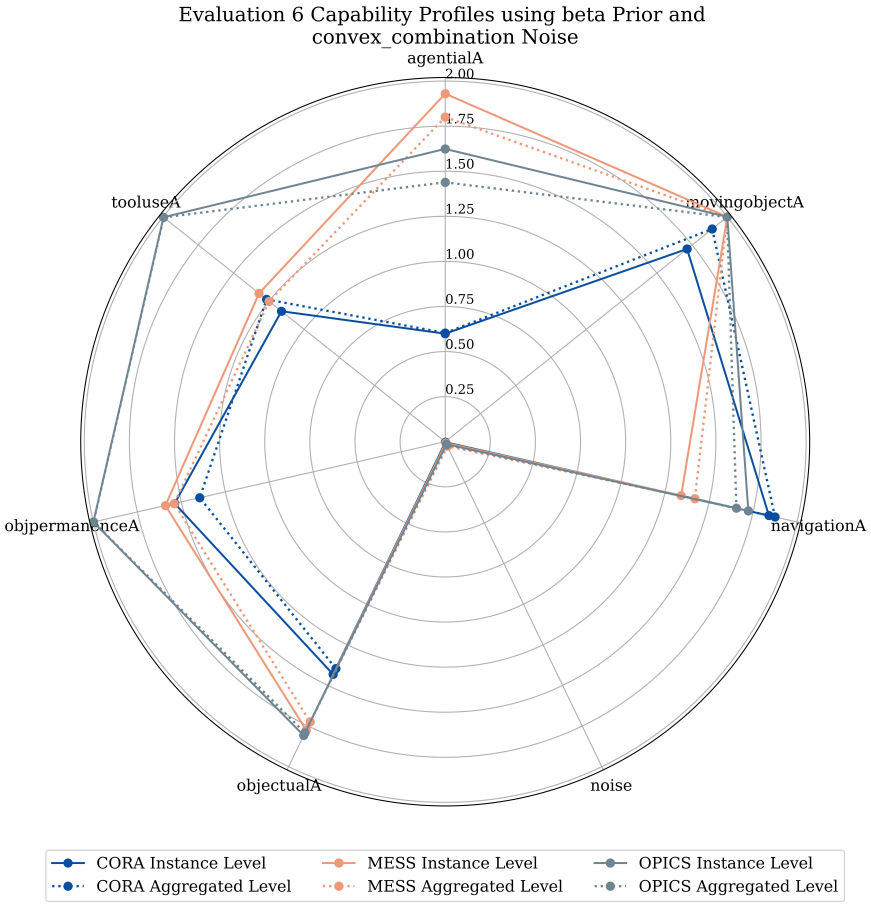

Figure 6.2: Radial Plot Capability Profiles from Evaluation 6 with Measurement Layouts Setting 2.

The noise term in this setting, influenced by a fixed average observed performance parameter, allows the model to balance observed data and inferred capabilities. In this case, the inferred noise value is practically zero for all agents and data granularities, which is a promising sign of the robustness of the measurement layouts and the variability of the evaluation data. This zero noise estimation suggests that the measurement layouts does not need to rely heavily on average observed performance to produce accurate forecasts about the agents' performance.

The predictive performance of the Measurement Layout, as shown in Tables 6.3 and 6.4, consistently outperforms simple aggregate estimation methods and even surpasses XGBoost at the instance level again. However, is still obtains slightly worse estimations at the aggregated level.

6.4 Evaluation 7: Measurement Layouts Setting 1

When analysing the approximated capability profiles using "Setting 1" with the Evaluation 7 data, the first noticeable pattern is the significant decline in Agential abilities for MESS, as shown in the radial plot in Figure 6.3. Indeed, from Evaluation 6 to Evaluation 7, MESS transitions from being the most capable agent in this facet to being the "weakest."

Agent	Performance	Aggregate (BS)	Measurement Layouts (BS)	Assessor (BS)
CORA	0.5716	0.2015	**0.2008**	0.2823
MESS	0.7589	0.2128	**0.1741**	0.2153
OPICS	0.8987	0.1692	**0.0862**	0.0970
Mean	0.7385	0.1721	**0.1537**	0.1980

Table 6.3: Predictive Performance from Measurement Layouts Setting 2 on Evaluation 6 - Instance Level.

Agent	Performance	Aggregate (BS)	Measurement Layouts (BS)	Assessor (BS)
CORA	0.5623	0.1313	0.0668	**0.0609**
MESS	0.7598	0.0855	0.0785	**0.0522**
OPICS	0.8935	0.0342	0.0298	**0.0224**
Mean	0.7385	0.0837	0.0583	**0.0440**

Table 6.4: Predictive Performance (Brier Score) from Measurement Layouts Setting 2 on Evaluation 6 - Aggregated Level.

Additionally, the Moving Object ability appears to have "improved" for all agents from one evaluation to the next. Furthermore, the capabilities of CORA seem to be "better" in this evaluation across almost every ability. Most significantly, CORA has transformed from lacking Agential capabilities, as represented in Figure 6.1, to being closely matched with OPICS. It is now the most capable agent regarding Tool Use ability and closely matches the other agents in terms of Object Permanence and Navigation capabilities.

This considerable improvement in CORA from Evaluation 6 to 7, as indicated by the changes in the radial plot of its cognitive profile, correlates with its relative performance improvement in this evaluation, confirmed by Tables 6.5 and 6.6. Indeed, the relative ordering of performance now places OPICS first, followed by CORA, and then MESS.

However, the difference between CORA and MESS is not particularly significant. It is also worth noting that despite a general drop in performance among the agents –except for MESS– the relative ordering of capabilities across the cognitive profile elements remains majorly unaltered from one evaluation to the next –with the exception of tool use ability. This stability suggests that the core "strengths" and "weaknesses" –from the capabilities perspective– of each agent are consistent, even if their overall performance fluctuates between evaluations.

Agent	Performance	Aggregate (BS)	Measurement Layouts (BS)	Assessor (BS)
CORA	0.7278	0.2015	**0.1967**	0.2341
MESS	0.6921	0.2128	**0.1929**	0.2431
OPICS	0.7902	0.1692	**0.1374**	0.1793
Mean	0.7367	0.1945	**0.1754**	0.2188

Table 6.5: Predictive Performance (Brier Score) from Measurement Layouts Setting 1 on Evaluation 7 - Instance Level.

The predictive performance of this first setting of the Measurement Layouts in Evaluation 7 is detailed in Tables 6.5 and 6.6, which summarise the performance metrics at the instance and aggregated levels, respectively. We can highlight the following observations:

At the instance level (Table 6.5), the Measurement Layouts shows again consistent improvement over the Aggregate baseline performance estimation for all agents. In terms of the aggregated level –Table 6.6–, the Measurement Layouts again performs better than

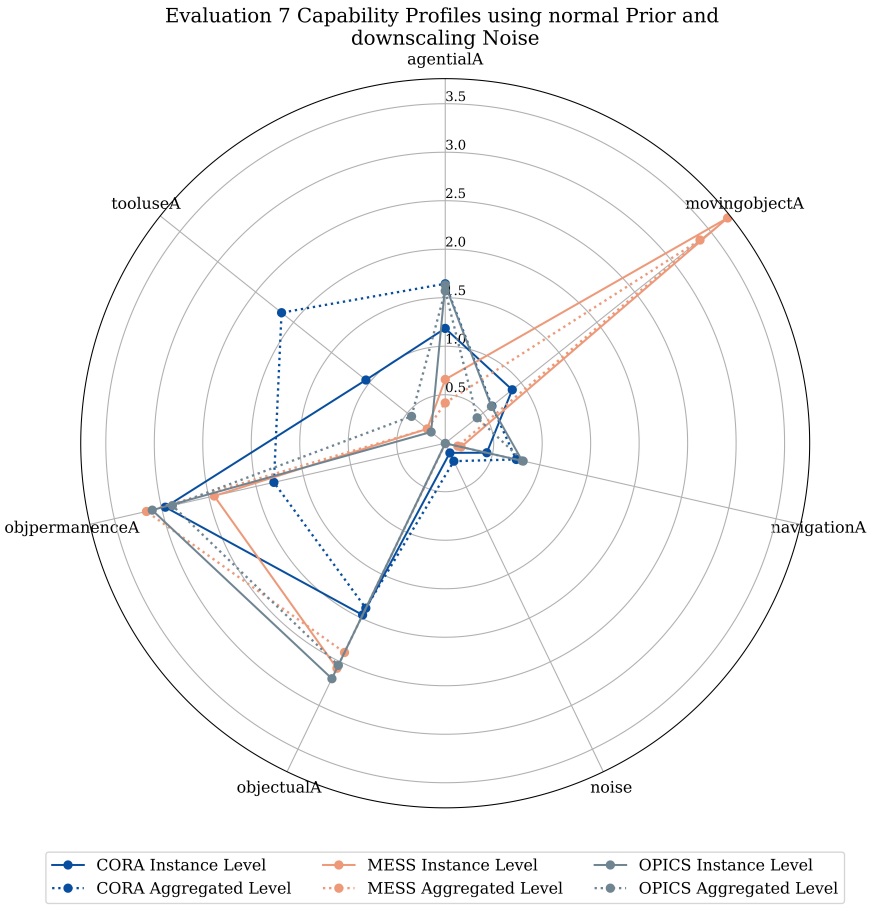

Figure 6.3: Radial Plot Capability Profiles from Evaluation 7 with Measurement Layouts Setting 1.

Agent	Performance	Aggregate (BS)	Measurement Layouts (BS)	Assessor (BS)
CORA	0.7069	0.0671	0.0672	**0.0498**
MESS	0.6907	0.0933	0.0716	**0.0497**
OPICS	0.7899	0.0660	**0.0504**	0.0589
Mean	0.7298	0.0756	0.0631	**0.0528**

Table 6.6: Predictive Performance (Brier Score) from Measurement Layouts Setting 1 on Evaluation 7 - Aggregated Level.

the Aggregate method, with lower Brier Scores across the board. It is noticeable that for CORA and MESS, the Brier Scores are very close to the Assessor scores. This "closeness" indicates that the measurement layouts is robust in predicting aggregated performance. This robustness is also proved by the fact that the noise estimations for every agent and considering instance and aggregated level data are consistently very close to 0. Taking into consideration the noise implementation of this setting, this result reinforces one of the Measurement Layout's strengths: that a complete well predefined set of cognitive profile abilities, coupled with diverse evaluation data, can reliably infer the agents' capabilities and achieve strong predictive performance.

6.5 Evaluation 7: Measurement Layouts Setting 2

The results inferred profiles for Evaluation 7 data with the second setting can be seen below in Figure 6.4. Just as a reminder, when using scaled beta priors, capabilities are bounded between 0 and 2 on an ordinal scale, which enables relative ordering but does not allow us for direct quantification of differences.

In this case, there are substantial changes from one evaluation to other in the approximated capability profiles. Nonetheless, taking into account the variations we observed using the setting 1 for this evaluation data, we can conclude that they are consistent. As an example, we see that using this second setting, CORA seems to be the most capable agent from the perspective of Tool Use ability, when it used to be the "weakest" from this perspective on Evaluation 6. However, this is also observed when using normal priors on Evaluation 7. This leads us to think that CORA's tool Use abilities could have been "enhanced" between these two evaluations.

Furthermore, similar to the observations with Setting 1, CORA and MESS have managed to match OPICS's Object Permanence capabilities. This capability was a key differentiator in Evaluation 6, contributing to OPICS's superior performance. This alignment of capabilities might explain the lack of significant performance differences between the agents in Evaluation 7, as reflected in the "Performance" columns in Tables 6.7 and 6.8.

Additionally, in relation to CORA's inferred profile, we observe notable differences between the instance and aggregated level capabilities, particularly in Tool Use, Object Permanence, and Moving Object abilities. This is evident in both settings tested for this evaluation, as illustrated in the radial plots in Figures 6.3 for setting 1 and 6.4 for setting 2. In relation to this, we observe that noise values for CORA are significantly different from zero, specially in comparison to the other agents. This suggests that measurement layouts inferences for CORA's performance are less "stable" and more dependent on the observed performance. This might be the reason why there is less consistency between instance and aggregated level inferred capability profiles.

Agent	Performance	Aggregate (BS)	Measurement Layouts (BS)	Assessor (BS)
CORA	0.7278	0.2015	**0.1975**	0.2341
MESS	0.6921	0.2128	**0.1968**	0.2431
OPICS	0.7902	0.1692	**0.1420**	0.1793
Mean	0.7367	0.1945	**0.1787**	0.2188

Table 6.7: Predictive Performance (Brier Score) from Measurement Layouts Setting 2 on Evaluation 7 - Instance Level.

When analysing the predictive performance of Measurement Layouts Setting 2 in Evaluation 7, we observe a similar trend of results as in previous settings and evaluations. Starting with the instance-level performance metrics as shown in Table 6.7, the

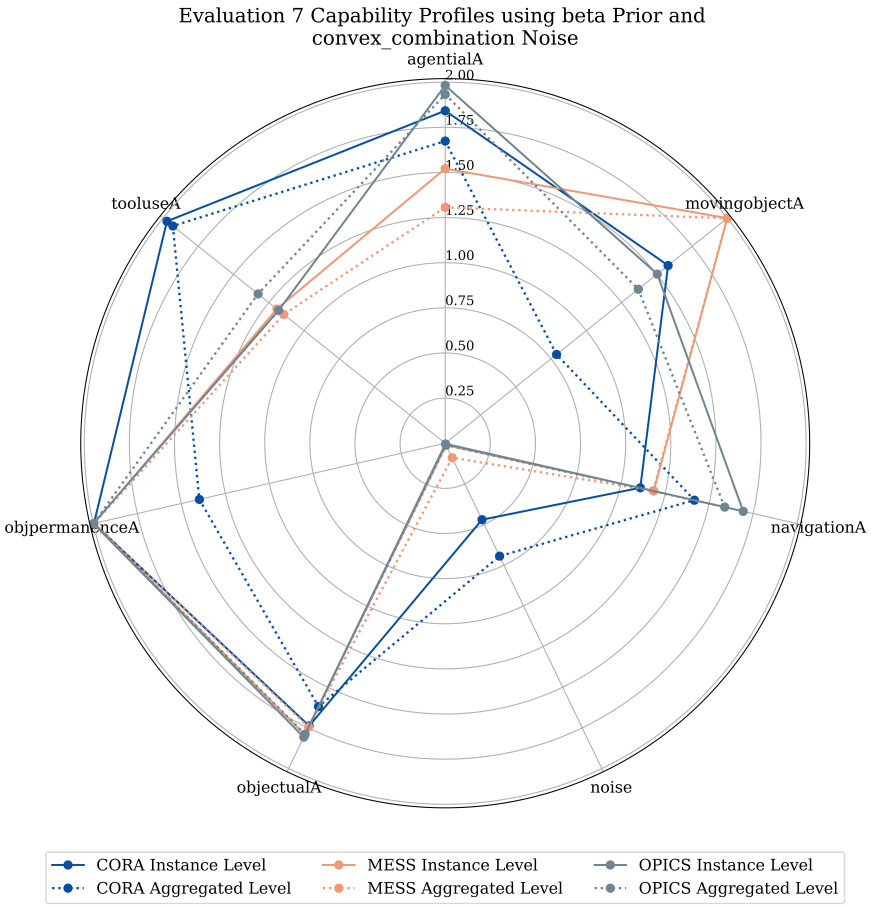

Figure 6.4: Radial Plot Capability Profiles from Evaluation 7 with Measurement Layouts Setting 2.

Agent	Performance	Aggregate (BS)	Measurement Layouts (BS)	Assessor (BS)
CORA	0.7069	0.0671	0.0669	**0.04978**
MESS	0.6907	0.0933	0.0765	**0.04967**
OPICS	0.7899	0.0663	**0.0545**	0.0589
Mean	0.7385	0.0756	0.066	**0.0528**

Table 6.8: Predictive Performance (Brier Score) from Measurement Layouts Setting 2 on Evaluation 7 - Aggregated Level

Measurement Layouts demonstrates consistent improvements over the Aggregate baseline performance estimations for all agents.

Additionally, at the aggregated level as detailed in Table 6.8, the Measurement Layouts again outperforms the Aggregate method. The lower Brier Scores indicate a more accurate prediction of aggregated performance across all agents. It's worth noting that the Measurement Layout's predictive performance is also close the XGBoost Assessor's.

6.6 Closing Remarks

- Taking a look back at the exploratory analysis about agents weaknesses and strengths carried out in Section 4.3, we concluded that for agents MESS and OPICS, abilities considered in the measurement layouts' cognitive profile to address Moving Object Reasoning, Core Object Reasoning and Object Permanence demands were expected to be advanced, at least in comparison to CORA. The inferred capability profiles from different measurement layouts' settings confirm this intuition. This demonstrates how the bottom-up inference of agents' capabilities can be effectively explained by the nuances of their behaviour. While these capabilities can be intuitively inferred from detailed performance data, the measurement layouts provide a comprehensive framework that robustly infers these capabilities integrating all performance data "at once" and demonstrates remarkable predictive power regarding the agents' future performance.

- The selection of different measurement layouts settings, specially with respect to the choice of priors, has allowed us to contrast the consistency of the measurement layouts when inferring agents' capability profiles independently of the hyperpameter choice. Opting for normal priors permitted us to determine the extent to which some agents seem more capable than others. On the other hand, beta priors provided an overview of the relative ordering of agents in terms of their capabilities.

- Measurement layouts framework demonstrates consistently stronger predictive power in relation to agents' performance for both Evaluation acts than the Assesor and the baseline prediction considering the instance level perspective. Also, despite predictions are more accurate at the aggregated level, XGBoost Assesor achieves slightly better predictability.

CHAPTER 7
Conclusions

7.1 Limitations and Future Work

As we have already mentioned multiple times previously, the fact that we have to rely on theoretical constructs from cognitive science for deciding the measurement layouts topology is a limiting factor. Measurement layouts are not as "simple" as common HBNs or other machine learning models, we do not have that much flexibility for "playing" with its inner structure and hyperparameters.

In the considered settings of the measurement layouts, it made sense to think that capabilities of agents could be compensatory (Definition3.3), i.e., high values for one ability can compensate for the weaker ones in a specific task. This is particularly evident when examining the inferred capability profiles of agents like MESS in setting 1 of Evaluation 6 (see Figure **??**). MESS demonstrates very advanced Moving Object and Agential Abilities, which can compensate for its weaker abilities in areas like Object Permanence and Navigation Ability.

Nonetheless, when we introduced compensation following the expression given by Equation 7.1, it resulted in a worse predictive performance from the measurement layout.

Then, we tried to find out another approach to model compensation. To achieve this, we proposed the following expression for integrating intermediate performance and therefore, modelling a compensatory setting:

$$1 - \prod_{l=1}^{L}(1 - \sigma_l * \alpha_l) \tag{7.1}$$

Here we took the original expression of compensatory performance, and add a weight –represented by $\alpha_l \sim Beta$– to each intermediate performance node. This weight represents to which extent a given intermediate performance node –which is at the end related to a specific ability– can compensate for the rest.

For example, taking our initial topology, if the weight $\alpha_{navigationA}$ inferred distribution mean is 1, we would expect navigation capabilities to fully compensate for the rest because in the case that navigation performance is 1, in that specific situation, the rest of capabilities would not matter, as final performance will be 1 independently of the rest intermediate performance values.

This approach of modelling compensation locally resulted to be promising, at least in comparison to the current approach. We compared the predictive performance –with multiple combinations of prior and noise initialisations–, and the new compensatory setting had more predictive power than the older one. This can be seen in Table C.1, available in the Appendix C.

For the Future Work, we propose:

- Keep exploring ways to model compensation.
- Study new topologies of the measurement layouts.
- Work on modelling capabilities distributions in such a way that they can be measured with a ratio scale.

7.2 Objectives Fulfilment

In relation to the specified objectives in Section 1.1, we can confirm the following:

- These objectives were tackled deeply throughout the results Section 6, having previously introduced the experimental procedure for testing the measurement layouts and the settings that we were going to consider in Section 5.

With respect to secondary objectives:

- The first was widely covered when presenting the potential of Bayesian Modelling in Cognitive Evaluation of Artificial Intelligence systems in Section 2.4.
- We provided a comprehensive view of the current state of AI Evaluation in Section 2, remarking its crucial role on the field of AI Safety and we also advocated for a change in the current paradigm of AI Evaluation, specially for assessing general-purpose systems. This covers the two first secondary objectives.
- In Section 2.4 we discussed the possibilities of the cognitive evaluation of AI, and how it may respond to the deficits of current evaluation approaches, covering the third objective.
- Finally, in Section 3.2.3 we introduced the role of Monte Carlo method on Bayesian statistics, commenting different existing variations of it, and we framed its implication on the measurement layouts framework later in Section 3.3.

7.3 Integration of Bachelor's Degree Competences

The present project represent the last stage of 4 years of continuous learning in the academic and personal scope. Therefore, I will highlight some specific courses that have been crucial to set the foundations to develop this work:

1. Linear Algebra (13998), Exploratory Data Analysis (14004), Statistical models for decision making I and II (14005 and 14006) and Descriptive and predictive models I and II (14010 and 14011): because these have represented the building blocks in probability theory (specially Bayesian Theory) and algebra for understanding the complexities of Hierarchical Bayesian Networks and approximate inference.
2. Programming (14003), Data Structures (14008) and Algorithmics (14007): because they helped me to acquire the fundamental coding and logic skills for implementing in Python the complex framework the measurement layouts are in the computational and coding scope. They also provided me with the guidelines to write clean code.

3. Descriptive and predictive models I and II (14010 and 14011) and Model evaluation, deployment and monitoring (14028): because they introduced me pivotal concepts from the Machine Learning field. Specially, I highlight the introduction and discussion of model evaluation practices. This discussion encouraged me to be interested on AI Evaluation and Safety.

Bibliography

[1] Ekin Akyürek et al. *What learning algorithm is in-context learning? Investigations with linear models*. 2023. arXiv: 2211.15661.

[2] Usman Anwar et al. "Foundational challenges in assuring alignment and safety of large language models". In: *arXiv preprint arXiv:2404.09932* (2024).

[3] Adrià Puigdomènech Badia et al. *Agent57: Outperforming the Atari Human Benchmark*. 2020. arXiv: 2003.13350.

[4] Gail Blattenberger and Frank Lad. "Separating the Brier score into calibration and refinement components: A graphical exposition". In: *The American Statistician* 39.1 (1985), pp. 26–32.

[5] Rishi Bommasani et al. *On the Opportunities and Risks of Foundation Models*. 2022. arXiv: 2108.07258 [cs.LG].

[6] Nick Bostrom. *Superintelligence*. Dunod, 2017.

[7] Nick Bostrom. *Superintelligence: Paths, strategies, dangers*. 2014.

[8] Nick Bostrom and Eliezer Yudkowsky. "The ethics of artificial intelligence". In: *Artificial intelligence safety and security*. Chapman and Hall/CRC, 2018, pp. 57–69.

[9] Will Bridewell and Paul Bello. "A theory of attention for cognitive systems". In: *Advances in Cognitive Systems* 4.1 (2016), pp. 1–16.

[10] Tom B. Brown et al. "Language Models are Few-Shot Learners". In: (2020). arXiv: 2005.14165 [cs.CL].

[11] J. Burden et al. *Measurement Layouts for Capability-Oriented AI Evaluation*. AAAI Tutorial. 2024.

[12] John Burden et al. *Inferring Capabilities from Task Performance with Bayesian Triangulation*. 2023. arXiv: 2309.11975 [cs.AI].

[13] Ryan Burnell et al. "Rethink reporting of evaluation results in AI". In: *Science* 380.6641 (2023), pp. 136–138. DOI: 10.1126/science.adf6369. eprint: https://www.science.org/doi/pdf/10.1126/science.adf6369. URL: https://www.science.org/doi/abs/10.1126/science.adf6369.

[14] Ryan Burnell et al. *Revealing the structure of language model capabilities*. 2023. arXiv: 2306.10062.

[15] John Bissell Carroll. *Human cognitive abilities: A survey of factor-analytic studies*. 1. Cambridge university press, 1993.

[16] Nick Chater, Joshua B Tenenbaum, and Alan Yuille. "Probabilistic models of cognition: Conceptual foundations". In: *Trends in cognitive sciences* 10.7 (2006), pp. 287–291.

[17] Tianqi Chen and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, Aug. 2016. DOI: 10.1145/2939672.2939785. URL: http://dx.doi.org/10.1145/2939672.2939785.

[18] Paul Christiano. "Clarifying "AI Alignment"". In: *Medium: AI Alignment* (2017).

[19] Zheng Chu et al. *Navigate through Enigmatic Labyrinth A Survey of Chain of Thought Reasoning: Advances, Frontiers and Future*. 2024. arXiv: 2309.15402.

[20] Allan Dafoe. "AI governance: a research agenda". In: *Governance of AI Program, Future of Humanity Institute, University of Oxford: Oxford, UK* 1442 (2018), p. 1443.

[21] David "davidad" Dalrymple et al. *Towards Guaranteed Safe AI: A Framework for Ensuring Robust and Reliable AI Systems*. 2024. arXiv: 2405.06624 [cs.AI].

[22] Qingxiu Dong et al. "A survey on in-context learning". In: *arXiv preprint arXiv:2301.00234* (2022).

[23] Simon Duane et al. "Hybrid monte carlo". In: *Physics letters B* 195.2 (1987), pp. 216–222.

[24] Ward Edwards, Harold Lindman, and Leonard J Savage. "Bayesian statistical inference for psychological research." In: *Psychological review* 70.3 (1963), p. 193.

[25] Deep Ganguli et al. "Predictability and Surprise in Large Generative Models". In: *2022 ACM Conference on Fairness, Accountability, and Transparency*. FAccT '22. ACM, June 2022. DOI: 10.1145/3531146.3533229. URL: http://dx.doi.org/10.1145/3531146.3533229.

[26] Katja Grace et al. *When Will AI Exceed Human Performance? Evidence from AI Experts*. 2018. arXiv: 1705.08807.

[27] David Gunning. *Machine Common Sense Concept Paper*. 2018. arXiv: 1810.07528.

[28] David Gunning and Proposers Day. "Machine Common Sense". In: *artificial intelligence* 58 (2015), pp. 92–103.

[29] Remco Heesen, Liam Kofi Bright, and Andrew Zucker. "Vindicating methodological triangulation". In: *Synthese* 196 (2019), pp. 3067–3081.

[30] Dan Hendrycks, Mantas Mazeika, and Thomas Woodside. "An overview of catastrophic ai risks". In: *arXiv preprint arXiv:2306.12001* (2023).

[31] Jose Hernandez-Orallo. "AI evaluation: On broken yardsticks and measurement scales". In: 2020.

[32] José Hernández-Orallo. *The Measure of All Minds: Evaluating Natural and Artificial Intelligence*. Cambridge University Press, 2017.

[33] José Hernández-Orallo, David L Dowe, and M Victoria Hernández-Lloreda. "Universal psychometrics: Measuring cognitive abilities in the machine kingdom". In: *Cognitive Systems Research* 27 (2014), pp. 50–74.

[34] José Hernández-Orallo and Karina Vold. "AI Extenders: The Ethical and Societal Implications of Humans Cognitively Extended by AI". In: *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. AIES '19. Honolulu, HI, USA: Association for Computing Machinery, 2019, pp. 507–513. ISBN: 9781450363242. DOI: 10.1145/3306618.3314238. URL: https://doi.org/10.1145/3306618.3314238.

[35] José Hernández-Orallo et al. "General intelligence disentangled via a generality metric for natural and artificial intelligence". In: *Scientific reports* 11.1 (2021), p. 22822.

[36] Matthew D Hoffman, Andrew Gelman, et al. "The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo." In: *J. Mach. Learn. Res.* 15.1 (2014), pp. 1593–1623.

[37] Xuefei Huang et al. "Demand Response Management for Industrial Facilities: A Deep Reinforcement Learning Approach". In: *IEEE Access* 7 (2019), pp. 82194–82205. DOI: 10.1109/ACCESS.2019.2924030.

[38] Ron S Kenett. "Bayesian networks: Theory, applications and sensitivity issues". In: *Encyclopedia with Semantic Computing and Robotic Intelligence* 1.01 (2017), p. 1630014.

[39] "Level of MeasurementLevel of measurement". In: *Encyclopedia of Public Health*. Ed. by Wilhelm Kirch. Dordrecht: Springer Netherlands, 2008, pp. 851–852. ISBN: 978-1-4020-5614-7. DOI: 10.1007/978-1-4020-5614-7_1971. URL: https://doi.org/10.1007/978-1-4020-5614-7_1971.

[40] Theodoros A Kyriazos et al. "Applied psychometrics: sample size and sample power considerations in factor analysis (EFA, CFA) and SEM in general". In: *Psychology* 9.08 (2018), p. 2207.

[41] Michael D Lee. "How cognitive modeling can benefit from hierarchical Bayesian models". In: *Journal of Mathematical Psychology* 55.1 (2011), pp. 1–7.

[42] Michael D Lee. "Three case studies in the Bayesian analysis of cognitive models". In: *Psychonomic Bulletin & Review* 15 (2008), pp. 1–15.

[43] Yuxi Li. *Deep Reinforcement Learning: An Overview*. 2018. arXiv: 1701.07274.

[44] Fernando Martínez-Plumed et al. "Research community dynamics behind popular AI benchmarks". In: *Nature Machine Intelligence* 3.7 (2021), pp. 581–589.

[45] Fernando Martínez-Plumed et al. "When ai difficulty is easy: The explanatory power of predicting irt difficulty". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 7. 2022, pp. 7719–7727.

[46] Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. *Circuit Component Reuse Across Tasks in Transformer Language Models*. 2024. arXiv: 2310.08744.

[47] Nicholas Metropolis et al. "Equation of state calculations by fast computing machines". In: *The journal of chemical physics* 21.6 (1953), pp. 1087–1092.

[48] Kevin P Murphy. *Probabilistic machine learning: an introduction*. MIT press, 2022.

[49] Zhaoyang Niu, Guoqiang Zhong, and Hui Yu. "A review on the attention mechanism of deep learning". In: *Neurocomputing* 452 (2021), pp. 48–62.

[50] INGMAR PERSSON and JULIAN SAVULESCU. "The Perils of Cognitive Enhancement and the Urgent Imperative to Enhance the Moral Character of Humanity". In: *Journal of Applied Philosophy* 25.3 (2008), pp. 162–177. DOI: https://doi.org/10.1111/j.1468-5930.2008.00410.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1468-5930.2008.00410.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1468-5930.2008.00410.x.

[51] Alec Radford et al. "Improving language understanding by generative pre-training". In: (2018).

[52] Alec Radford et al. "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8 (2019), p. 9.

[53] Ibraheem Rehman et al. "Classical conditioning". In: (2017).

[54] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Pearson, 2016.

[55] John Rust and Susan Golombok. *Modern psychometrics: The science of psychological assessment*. Routledge, 2014.

[56] John Salvatier, Thomas Wiecki, and Christopher Fonnesbeck. *Probabilistic Programming in Python using PyMC*. 2015. arXiv: 1507.08050.

[57] David Schlangen. “Language tasks and language games: On methodology in current natural language processing research”. In: *arXiv preprint arXiv:1908.10747* (2019).

[58] Toby Shevlane et al. *Model evaluation for extreme risks*. 2023. arXiv: 2305.15324 [cs.AI].

[59] David Silver et al. “Mastering the game of Go with deep neural networks and tree search”. In: *nature* 529.7587 (2016), pp. 484–489.

[60] Charles Spearman. “"General Intelligence" Objectively Determined and Measured.” In: (1961).

[61] Aarohi Srivastava et al. *Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models*. 2023. arXiv: 2206.04615 [cs.CL].

[62] S. S. Stevens. “On the Theory of Scales of Measurement”. In: *Science* 103.2684 (1946), pp. 677–680. DOI: 10.1126/science.103.2684.677. eprint: https://www.science.org/doi/pdf/10.1126/science.103.2684.677. URL: https://www.science.org/doi/abs/10.1126/science.103.2684.677.

[63] Elizabeth Spelke Susan E. Carey. “Science and Core Knowledge”. In: *Philosophy of Science* 63.4 (1996), pp. 515–533.

[64] Gemini Team et al. “Gemini: A Family of Highly Capable Multimodal Models”. In: (2024). arXiv: 2312.11805 [cs.CL].

[65] Max Tegmark. *Life 3.0: Being Human in the Age of Artificial Intelligence*. Knopf Publishing Group, 2017. ISBN: 1101946598.

[66] Georgios Tsaparlis. “Cognitive Demand”. In: *Encyclopedia of Science Education*. Ed. by Richard Gunstone. Dordrecht: Springer Netherlands, 2021, pp. 1–4. ISBN: 978-94-007-6165-0. DOI: 10.1007/978-94-007-6165-0_40-20. URL: https://doi.org/10.1007/978-94-007-6165-0_40-20.

[67] A. M. Turing. “On Computable Numbers, with an Application to the Entscheidungsproblem”. In: *Proceedings of the London Mathematical Society* s2-42.1 (1937), pp. 230–265. DOI: https://doi.org/10.1112/plms/s2-42.1.230. eprint: https://londmathsoc.onlinelibrary.wiley.com/doi/pdf/10.1112/plms/s2-42.1.230. URL: https://londmathsoc.onlinelibrary.wiley.com/doi/abs/10.1112/plms/s2-42.1.230.

[68] Jodie B Ullman and Peter M Bentler. “Structural equation modeling”. In: *Handbook of Psychology, Second Edition* 2 (2012).

[69] Ashish Vaswani et al. “Attention Is All You Need”. In: (2023). arXiv: 1706.03762 [cs.CL].

[70] Alex Wang et al. *GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding*. 2019. arXiv: 1804.07461.

[71] Xiting Wang et al. *Evaluating General-Purpose AI with Psychometrics*. 2023. arXiv: 2310.16379 [cs.AI].

[72] Jason Wei et al. *Emergent Abilities of Large Language Models*. 2022. arXiv: 2206.07682 [cs.CL].

[73] Wikipedia contributors. *Commonsense reasoning — Wikipedia, The Free Encyclopedia*. [Online; accessed 17-June-2024]. 2024. URL: https://en.wikipedia.org/w/index.php?title=Commonsense_reasoning&oldid=1213909823.

[74] Roman V Yampolskiy. “Predicting future AI failures from historic examples”. In: *foresight* 21.1 (2019), pp. 138–152.

[75] Roman V. Yampolskiy. *The Singularity May Be Near*. 2017. arXiv: 1706.01303 [cs.AI].

[76] Zhengyuan Yang et al. *The Dawn of LMMs: Preliminary Explorations with GPT-4V(ision)*. 2023. arXiv: 2309.17421 [cs.CV].

APPENDIX A

Appendix A: Exploratory Analysis Figures

A.1 Evaluation 6 Exploratory Analysis

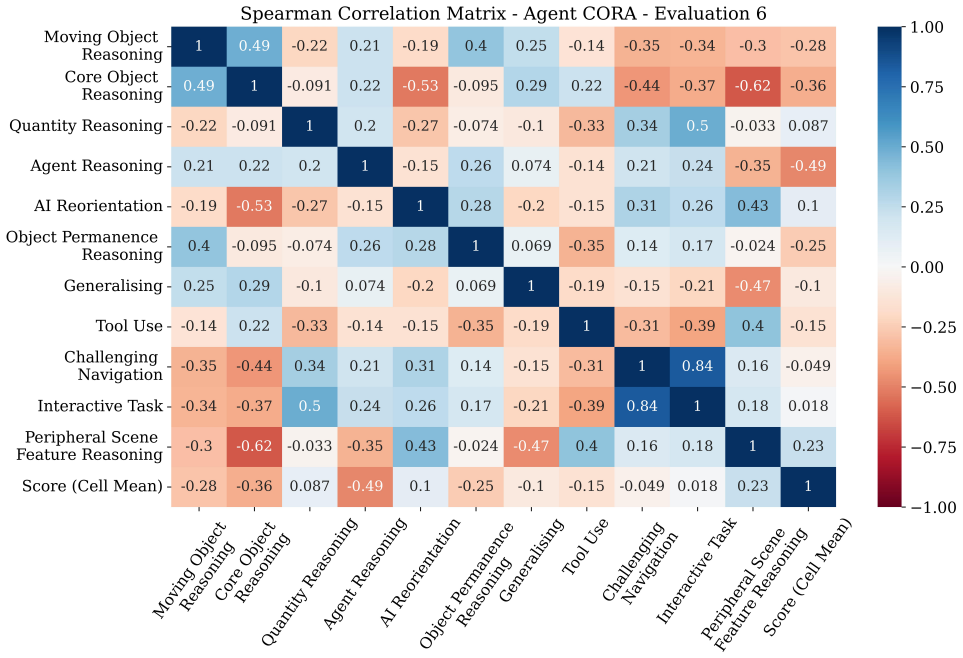

Figure A.1: Spearman's Correlation Heatmap for Agent CORA in Evaluation 6

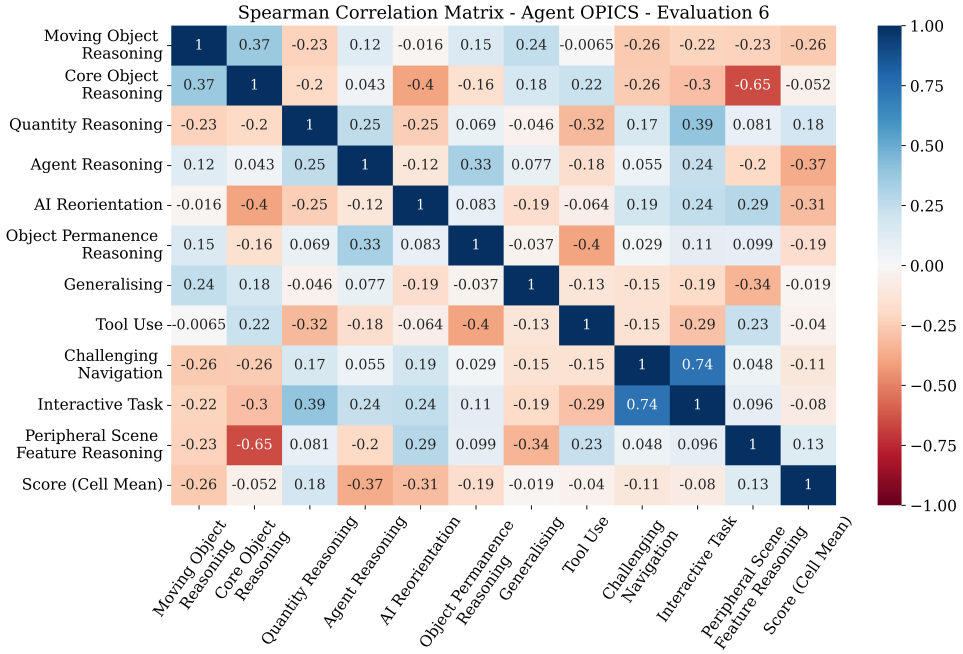

Figure A.2: Spearman's Correlation Heatmap for Agent MESS in Evaluation 6

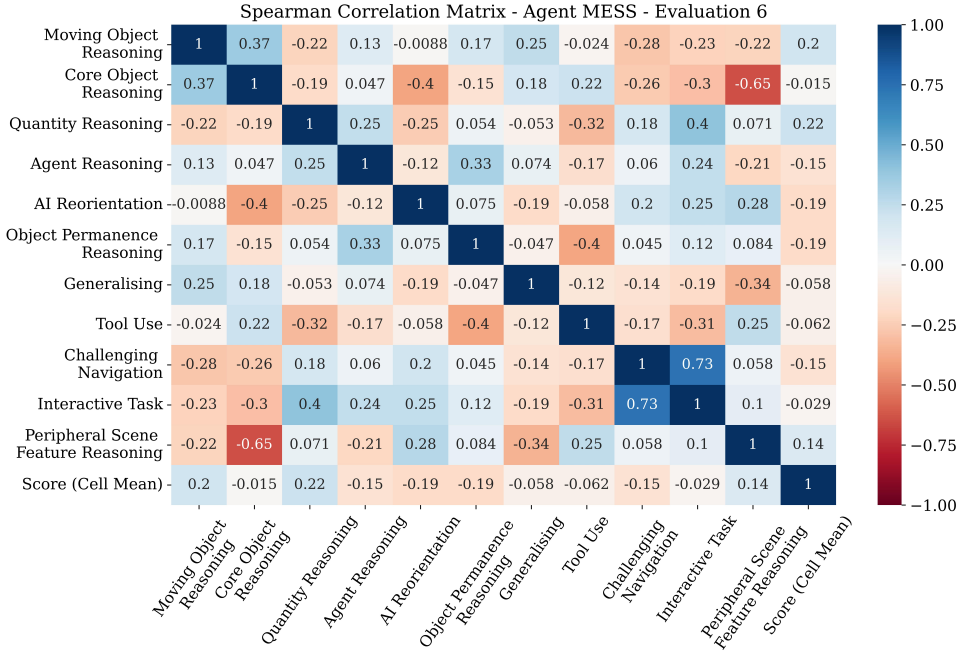

Figure A.3: Spearman's Correlation Heatmap for Agent OPICS in Evaluation 6

A.2 Evaluation 7 Exploratory Analysis

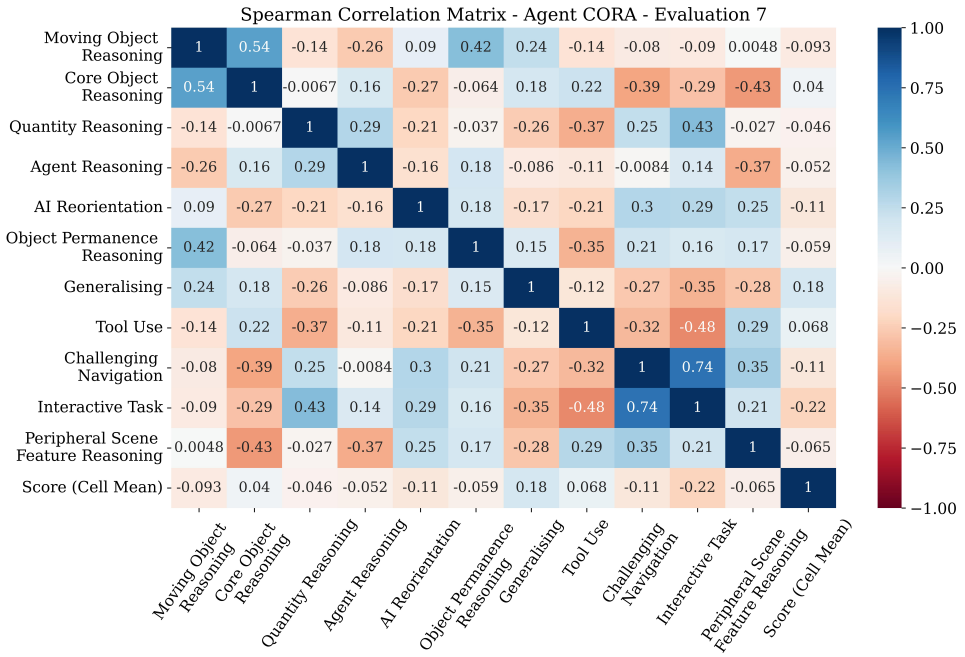

Figure A.4: Spearman's Correlation Heatmap for Agent CORA in Evaluation 7

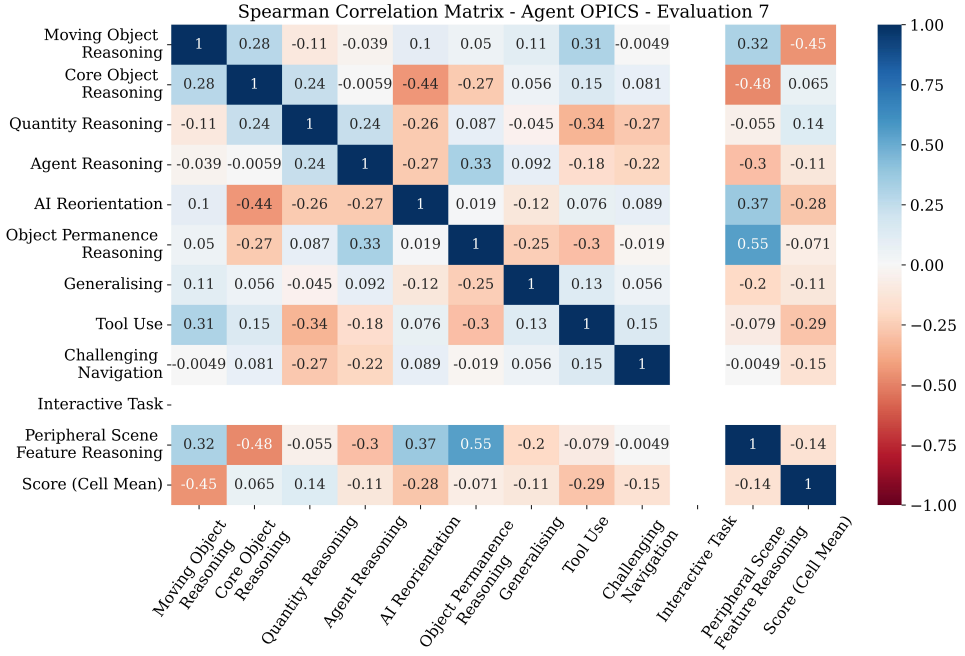

Figure A.5: Spearman's Correlation Heatmap for Agent MESS in Evaluation 7

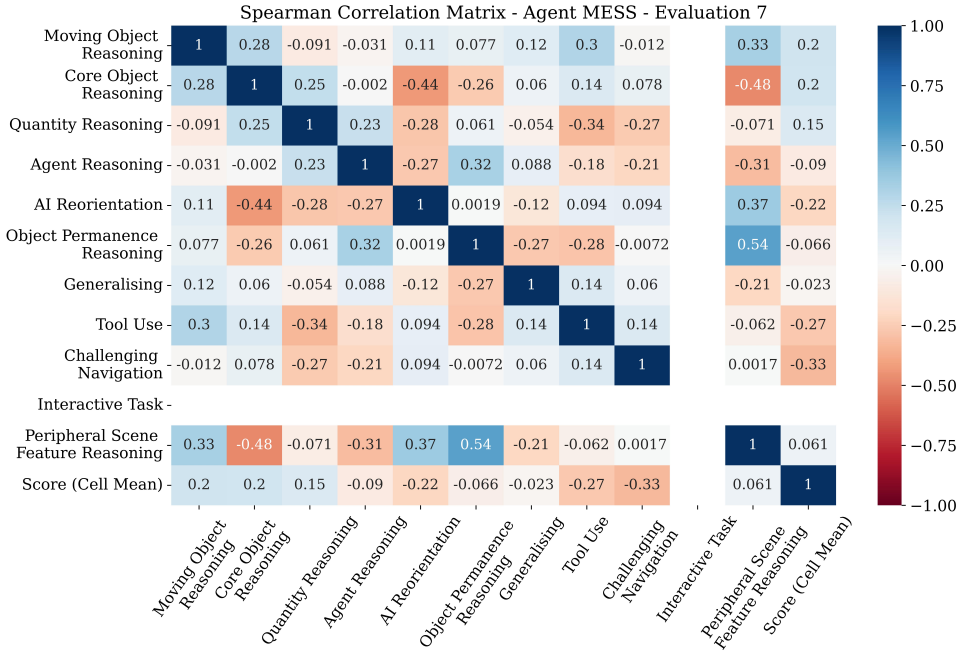

Figure A.6: Spearman's Correlation Heatmap for Agent OPICS in Evaluation 7

APPENDIX B

Appendix B: Capability Profiles Table Summaries

Agent	Ability	Ability_mean		Ability_sd	
		Instance Level	Aggregated Level	Instance Level	Aggregated Level
CORA	agentialA	-0.78	-0.75	0.05	0.08
	movingobjectA	0.80	1.32	0.08	0.46
	navigationA	0.58	0.97	0.05	0.37
	noise	0.008	0.05	0.007	0.02
	objectualA	0.53	0.53	0.04	0.08
	objpermanenceA	0.84	0.57	0.06	0.13
	tooluseA	0.16	0.28	0.06	0.13
MESS	agentialA	1.98	1.47	0.41	0.43
	movingobjectA	3.22	2.78	0.47	0.51
	navigationA	0.60	0.78	0.03	0.10
	noise	0.003	0.008	0.003	0.01
	objectualA	1.18	1.13	0.005	0.09
	objpermanenceA	0.74	0.66	0.03	0.05
	tooluseA	0.44	0.34	0.02	0.04
OPICS	agentialA	0.77	0.56	0.04	0.08
	movingobjectA	1.19	1.09	0.04	0.06
	navigationA	0.89	1.29	0.03	0.22
	noise	0.001	0.001	0.001	0.001
	objectualA	2.47	2.55	0.27	0.24
	objpermanenceA	3.39	2.54	0.64	0.64
	tooluseA	1.03	1.02	0.04	0.07

Table B.1: Capability Profiles from Evaluation 6 and Measurement Layouts Setting 1

Agent	Ability	Ability_mean		Ability_sd	
		Aggregated Level	Instance Level	Aggregated Level	Instance Level
CORA	agentialA	0.602	0.599	0.053	0.027
	movingobjectA	1.893	1.714	0.110	0.104
	navigationA	1.875	1.840	0.096	0.098
	noise	0.010	0.003	0.009	0.003
	objectualA	1.396	2.012	0.049	0.036
	objpermanenceA	1.395	1.538	0.066	0.043
	tooluseA	1.266	1.161	0.108	0.044
MESS	agentialA	1.801	1.930	0.118	0.051
	movingobjectA	1.965	1.997	0.005	0.002
	navigationA	1.419	1.341	0.056	0.022
	noise	0.027	0.008	0.028	0.008
	objectualA	1.725	1.775	0.035	0.018
	objpermanenceA	1.541	1.593	0.049	0.017
	tooluseA	1.249	1.320	0.033	0.020
OPICS	agentialA	1.439	1.624	0.049	0.038
	movingobjectA	1.998	1.997	0.010	0.002
	navigationA	1.655	1.738	0.093	0.032
	noise	0.019	0.015	0.019	0.015
	objectualA	1.792	1.881	0.012	0.010
	objpermanenceA	1.999	2.000	0.004	0.002
	tooluseA	1.996	1.997	0.018	0.014

Table B.2: Capability Profiles from Evaluation 6 and Measurement Layouts Setting 2

Agent	Ability	Ability_mean		Ability_sd	
		Aggregated Level	Instance Level	Aggregated Level	Instance Level
CORA	agentialA	1.644	1.184	0.508	0.228
	movingobjectA	0.616	0.884	0.084	0.058
	navigationA	0.750	0.440	0.265	0.042
	noise	0.205	0.110	0.032	0.011
	objectualA	1.887	1.965	0.342	0.304
	objpermanenceA	1.812	2.961	0.764	0.638
	tooluseA	2.159	1.046	0.802	0.110
MESS	agentialA	0.416	0.657	0.064	0.030
	movingobjectA	3.361	3.724	0.651	0.614
	navigationA	0.134	0.168	0.027	0.024
	noise	0.005	0.003	0.005	0.003
	objectualA	2.393	2.574	0.348	0.252
	objpermanenceA	3.160	2.443	0.703	0.373
	tooluseA	0.236	0.237	0.094	0.049
OPICS	agentialA	1.571	1.633	0.258	0.037
	movingobjectA	2.421	2.097	0.436	0.032
	navigationA	0.825	0.814	0.054	0.040
	noise	0.001	0.001	0.002	0.002
	objectualA	2.541	2.693	0.236	0.234
	objpermanenceA	2.886	3.099	0.785	0.768
	tooluseA	0.445	0.186	0.094	0.045

Table B.3: Capability Profiles from Evaluation 7 and Measurement Layouts Setting 1

Agent	Ability	Ability_mean		Ability_sd	
		Instance Level	Aggregated Level	Instance Level	Aggregated Level
CORA	agentialA	1.674	1.842	0.238	0.111
	movingobjectA	0.789	1.580	0.191	0.068
	navigationA	1.416	1.109	0.307	0.033
	noise	0.694	0.470	0.063	0.049
	objectualA	1.619	1.738	0.093	0.033
	objpermanenceA	1.398	1.997	0.310	0.012
	tooluseA	1.930	1.972	0.129	0.065
MESS	agentialA	1.307	1.521	0.054	0.038
	movingobjectA	1.998	2.000	0.006	0.003
	navigationA	1.182	1.185	0.027	0.014
	noise	0.089	0.012	0.067	0.012
	objectualA	1.745	1.792	0.031	0.011
	objpermanenceA	1.996	1.997	0.012	0.007
	tooluseA	1.144	1.193	0.076	0.036
OPICS	agentialA	1.933	1.982	0.057	0.017
	movingobjectA	1.368	1.503	0.031	0.023
	navigationA	1.589	1.694	0.044	0.038
	noise	0.011	0.004	0.011	0.005
	objectualA	1.789	1.807	0.014	0.010
	objpermanenceA	1.999	2.000	0.009	0.006
	tooluseA	1.327	1.181	0.064	0.029

Table B.4: Capability Profiles from Evaluation 7 and Measurement Layouts Setting 2

APPENDIX C

Appendix C: Comparison of Compensatory Settings

Algorithm	Prior and Noise Type	Brier Score **New** Compensatory Setting	Brier Score **Original** Compensatory Setting
CORA	Normal, convex combination	**0.062289**	0.066025
MESS	Normal, convex combination	**0.088896**	0.091138
OPICS	Normal, convex combination	**0.068348**	0.082667
CORA	Normal, downscaling	**0.071292**	0.078174
MESS	Normal, downscaling	**0.090687**	0.095992
OPICS	Normal, downscaling	0.070841	**0.062096**
CORA	Beta, convex combination	0.068163	**0.065815**
MESS	Beta, convex combination	**0.088994**	0.091112
OPICS	Beta, convex combination	0.068728	**0.056770**
CORA	Beta, downscaling	0.076020	**0.069088**
MESS	Beta, downscaling	**0.091389**	0.094273
OPICS	Beta, downscaling	0.071712	**0.061129**

Table C.1: Brier Scores for Different Algorithms and Settings.

APPENDIX D

Appendix D: Sustainable Development Goals

Sustainable Development Goals	High	Medium	Low	Not Applicable
SDG 1. **No Poverty.**				X
SDG 2. **Zero Hunger.**				X
SDG 3. **Good Health and Well-being.**				X
SDG 4. **Quality Education.**				X
SDG 5. **Gender Equality.**				X
SDG 6. **Clean Water and Sanitation.**				X
SDG 7. **Affordable and Clean Energy.**				X
SDG 8. **Decent Work and Economic Growth.**	X			
SDG 9. **Industry, Innovation and Infrastructure.**	X			
SDG 10. **Reduced Inequality.**		X		
SDG 11. **Sustainable Cities and Communities.**			X	
SDG 12. **Responsible Consumption and Production.**	X			
SDG 13. **Climate Action.**				X
SDG 14. **Life Below Water.**				X
SDG 15. **Life on Land.**				X
SDG 16. **Peace, Justice and Strong Institutions.**				X
SDG 17. **Partnerships for the Goals.**				X

The impact of this project on the Sustainable Development Goals (SDGs) can be discussed principally in relation to its potential contributions to innovation, industry and infrastructure; as well as decent work and economic growth, which we explicitly point out as areas where out project has a high potential impact. These areas correspond to the goals:

- **SDG 9. Industry, Innovation and Infrastructure**: The reason why our project has a high impact on this goal because it contributes to it by presenting a more sophisticated methodology for evaluating Artificial Intelligence systems capabilities. This innovative approach could be critical in the future for industries reliant on AI technologies, as it helps them understanding and predicting AI systems behaviour.
- **SDG 8. Decent Work and Economic Growth.**: the project promotes decent work by potentially improving how AI agents can be used in various sectors to support jobs and by fostering an economic environment benefited by reliable AI technologies. Better AI evaluations can lead to more effective and safer AI implementations, fostering growth and supporting employment in tech-driven sectors.
- **SDG 10. Reduced Inequality**: the project indirectly contributes to reduced inequality by enhancing the capability to evaluate AI systems that could be used in a variety of applications, potentially making advanced technology more accessible and beneficial across different demographics and sectors.
- **SDG 11. Sustainable Consumption and Production**: we note its impact on this goal as a low, reflecting the project's marginal but existing effects on urban sustainability directly but acknowledging that improved AI can play a role in smarter urban development indirectly through better infrastructure planning and management technologies.

ETS Enginyeria Informàtica
Camí de Vera, s/n, 46022, València
T +34 963 877 210
F +34 963 877 219
etsinf@upvnet.upv.es - www.inf.upv.es