

Article

Not peer-reviewed version

---

# Constrained Object Hierarchies as a Unified Theoretical Model for Intelligence and Intelligent Systems

---

[Harris Wang](#)\*

Posted Date: 24 September 2025

doi: 10.20944/preprints202509.2011.v1

Keywords: artificial general intelligence; cognitive architecture; neural-symbolic integration; intelligence formalization; COH framework; GISMOL toolkit



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

# Constrained Object Hierarchies as a Unified Theoretical Model for Intelligence and Intelligent Systems

Harris Wang

Athabasca University, Canada; harrisw@athabascau.ca; Tel.: +1-800-301-6337

## Abstract

Achieving Artificial General Intelligence (AGI) requires a unified framework capable of modeling the full spectrum of intelligence—from logical reasoning and sensory perception to emotional regulation and collective behavior. This paper introduces Constrained Object Hierarchies (COH), a neuroscience-inspired theoretical model that represents intelligent systems as hierarchical compositions of objects governed by symbolic structure, neural adaptation, and constraint-based control. Each object is defined by a 9-tuple  $O = (C, A, M, N, E, I, T, G, D)$ , encapsulating its Components, Attributes, Methods, Neural components, Embedding, and governing Identity, Trigger, Goal, and Daemon constraints. We demonstrate COH's expressiveness by formalizing 19 distinct intelligence types—including human-centric, artificial, and collective intelligences—each with detailed COH parameters and implementation blueprints. These formalizations span logical-mathematical, linguistic, spatial, emotional, social, computational, perceptual, motor, and embodied intelligences, among others. To bridge theory and practice, we introduce GISMOL, a Python-based toolkit for instantiating COH objects and executing their constraint systems and neural components. GISMOL enables modular development and integration of intelligent agents, supporting a structured methodology for AGI system design. By unifying symbolic and connectionist paradigms under a constraint-governed architecture, COH provides a scalable foundation for building general-purpose intelligent systems.

**Keywords:** artificial general intelligence; cognitive architecture; neural-symbolic integration; intelligence formalization; COH framework; GISMOL toolkit

## 1. Introduction

The field of artificial intelligence has historically oscillated between paradigms—symbolic AI, which excels at rule-based reasoning, and connectionist AI, which dominates pattern recognition [1]. A significant challenge on the path to AGI is the integration of these paradigms into a cohesive architecture that can exhibit the structural, adaptive, and constraint-based nature of biological intelligence [2]. Existing cognitive architectures, such as SOAR [3] and ACT-R [4], provide robust models for specific cognitive processes but often lack a flexible, unifying formalism for the vast spectrum of intelligence types described by psychological and computational theories [5,6].

This paper presents the **Constrained Object Hierarchies (COH)** framework, designed to address this integration challenge. COH is predicated on the neuroscientific principle of functional hierarchy and constrained computation [7,8]. It models any intelligent agent, from a single module to a full AGI, as a compositional hierarchy of objects, each defined by a precise 9-tuple. This formalism allows for the seamless combination of symbolic components (C, A, M) with neural adaptive elements (N, E), all under the governance of a rich constraint system (I, T, G, D) that ensures stability, triggers appropriate behaviors, and pursues goals.

The primary contribution of this work is the comprehensive formalization of 19 known intelligence types using the COH framework. This serves a dual purpose: it validates COH's expressiveness and provides a concrete blueprint for implementation. The subsequent sections are structured as follows: a review of related work, a detailed exposition of the COH framework, a systematic formalization of each intelligence type as a COH object, and a discussion on the implementation pathway via GISMOL.

## 2. Literature Review

The COH framework sits at the intersection of cognitive science, computer science, and neuroscience. Its design draws from several key areas of research.

**Cognitive Architectures:** Systems like SOAR [3] and ACT-R [4] are seminal works that model human cognition through production rules and memory chunks. While powerful, they can be complex to extend to novel domains or to integrate with modern deep learning. COH adopts a more flexible, object-oriented compositional approach, making it inherently modular and scalable.

**Neural-Symbolic Integration:** A central challenge for AGI is merging neural networks with symbolic reasoning [9,10]. COH explicitly allocates roles for both: symbolic components (C, A, M) handle structured knowledge and operations, while neural components (N, E) provide adaptive, sub-symbolic pattern recognition and representation learning [11,12].

**Theories of Intelligence:** COH's scope is informed by pluralistic theories of intelligence. Howard Gardner's theory of multiple intelligences [5] provides a foundation for the human-centric types (e.g., linguistic, spatial). For artificial and collective types, we draw from computational intelligence [13], swarm intelligence [14], and embodied cognition [15], which argues that intelligence emerges from the interaction between an agent and its environment.

**Hierarchical and Constrained Models:** The hierarchical nature of COH is supported by neuroscientific evidence of functional hierarchies in the brain [7]. The constraint-based governance (I, T, G, D) is inspired by models of cognitive control [16] and behavior regulation, where systems operate within fundamental invariants and continuously monitor their state to achieve goals.

### 3. The COH Framework: A Formal Definition

A Constrained Object Hierarchy (COH) is a formal system designed to represent a fundamental intelligent entity within an intelligent system. It is defined recursively, as every object is itself composed of sub-objects. The complete state of an intelligent agent is therefore a complex, interacting graph of COH instances. The fundamental unit, a single Constrained Object  $O$ , is formally represented as a 9-tuple:

$$O = (C, A, M, N, E, I, T, G, D)$$

The semantics of the 9-tuple is explained in section 3.1, section 3.2 and section 3.3.

#### 3.1. Structural and State Components (C, A, M)

These components define the object's structure, state, and available actions.

**C: Components.** This is a set of sub-objects that compose the current object.  $C = \{O_1, O_2, \dots, O_k\}$ . This establishes the compositional hierarchy. A primitive object (e.g., a sensory pixel) has  $C = \emptyset$ .

**A: Attributes.** This is a set of state variables that describe the object's current state.  $A = \{a_1: v_1, a_2: v_2, \dots, a_n: v_n\}$ , where  $v_i$  can be a scalar, vector, or symbolic value.

**M: Methods.** This is a set of executable procedures or actions that can be invoked to change the object's state or affect its environment.  $M = \{m_1, m_2, \dots, m_p\}$ . Each method  $m_i$  can be seen as a function that operates on  $A$  and possibly the attributes of objects in  $C$ .

### 3.2. Adaptive Components ( $N, E$ )

These components endow the object with the ability to learn and adapt, providing the sub-symbolic, continuous grounding for the symbolic structure.

**N: Neural Components.** This is a set of adaptive, typically parameterized models (e.g., neural networks, Bayesian filters) associated with the object.  $N = \{n_1(\theta_1), n_2(\theta_2), \dots\}$ . A component  $n_i$  could be responsible for predicting an attribute's value, classifying the object's state, or learning the preconditions for a method  $m_i$ .

**E: Embedding Neural Component.** This is a specific, often central, neural component that learns a dense vector representation (an embedding) for the entire object based on its state ( $A$ ), the state of its components ( $C$ ), and its context.  $e = E(A, A\_C, context; \theta\_E)$ . This embedding serves as a subsymbolic summary that can be used for similarity comparison, semantic reasoning, and as an input to other models in  $N$ .

### 3.3. Constraint System ( $I, T, G, D$ )

This is the core innovation of the COH model, defining the "rules of engagement" for the object.

**I: Identity Constraints.** These are invariant rules that define the object's fundamental nature and must always hold. They are expressed as first-order logic predicates over  $A$  and  $C$ .  $I = \{\forall t \phi_1(A(t), C(t)), \forall t \phi_2(A(t), C(t)), \dots\}$ . If an identity constraint is violated, the object is in an invalid or error state. For example, an object representing a "door" might have  $I = \{is\_open \in \{True, False\}\}$ .

**T: Trigger Constraints.** These are event-condition-action (ECA) rules that define reactive behavior.  $T = \{(event, condition, action)\}$ . The event can be a change in  $A$ , a method call, or a signal from a parent/child object. The condition is a predicate checked against the state. The action is a method (from  $M$  or a sub-object's  $M$ ) to be executed. This provides a formal mechanism for hard-coded or learned reflexes.

**G: Goal Constraints.** These are optimization objectives that the object should strive to achieve. They are expressed as functions to be maximized or minimized.  $G = \{maximize f_1(A), minimize f_2(A), \dots\}$ . These can be intrinsic (e.g., an agent's energy level) or extrinsic (assigned by a parent object). This aligns the object's behavior with overall system goals.

**D: Constraint Daemons.** These are continuous, background processes that monitor the constraint satisfaction levels. A daemon  $d_i \in D$  continuously evaluates a constraint (from  $I, T$ , or  $G$ ) and can activate methods or signal other objects if a constraint is nearing violation or if an optimization opportunity is detected.  $D$  ensures the constraints are active, dynamic forces, not passive statements.

### 3.4. Neuroscientific Correlates of the COH Model

The COH model is not merely an engineering construct; its components are designed to mirror organizational principles observed in the mammalian brain.

The hierarchical composition ( $C$ ) directly reflects the layered structure of the neocortex, particularly the feedforward and feedback pathways in the sensory and frontal hierarchies (Felleman & Van Essen, 1991). The **Attributes (A)** correspond to the persistent activity patterns observed in working memory (Goldman-Rakic, 1995). The **Methods (M)** find their correlate in the population codes of the motor cortex that generate specific actions (Georgopoulos et al., 1986).

Critically, the adaptive components  $N$  and  $E$  are modeled on the brain's capacity for predictive coding and representation learning. The **Embedding (E)** can be seen as the role of the hippocampus and surrounding medial temporal lobe structures in creating cognitive maps and relational representations (O'Keefe & Nadel, 1978). The **Neural Components (N)** collectively represent the vast repertoire of learned predictive models throughout the cortex (Bar, 2007).

The constraint system within the COH framework exhibits strong parallels with established neuroscientific structures. **Identity Constraints (I)** correspond to the ventral visual stream and conceptual networks, which are responsible for invariant object recognition and categorical stability



(DiCarlo et al., 2012). **Trigger Constraints (T)** reflect the role of the amygdala and basal ganglia in mediating rapid, conditioned responses to stimuli—essential for reflexive and adaptive behavior (LeDoux, 2000). **Goal Constraints (G)** align with the functions of the prefrontal cortex and anterior cingulate cortex, which are central to maintaining goal representations, evaluating outcomes, and guiding executive control (Miller & Cohen, 2001). Finally, **Constraint Daemons (D)** are analogous to the default mode network and thalamic regulatory systems, which continuously monitor internal states, maintain system homeostasis, and signal salient events to higher-order cognitive centers (Raichle, 2015).

This formalism offers a universal language for both deconstructing and designing intelligent systems. To demonstrate its unified expressive power, in the next two sections we formalize a comprehensive set of known intelligence types using the COH theoretical model. Detailed justifications for each formalization are provided in **Supplemental Material A**. Due to the modular and declarative nature of the COH framework, these formalizations can be implemented as executable intelligent systems using any modern programming language. In **Supplemental Material B**, we illustrate this process by presenting a full implementation of Swarm Intelligence using **GISMOL**, a Python-based toolkit being developed specifically to instantiate and execute COH objects.

## 4. Formalization of Human-Centric Intelligences

Having established the theoretical foundations of the COH framework, we now demonstrate its expressive power by applying it to formalize a suite of human-centric intelligences. Drawing primarily from Gardner's theory of multiple intelligences [5], this section deconstructs nine core faculties—from logical reasoning and linguistic prowess to emotional and social understanding—into their constituent COH parameters. Each formalization provides a concrete blueprint for how these diverse, often overlapping forms of human cognition can be systematically represented and implemented as a hierarchical, constrained object.

### 4.1. Logical-Mathematical Intelligence

This intelligence involves deductive and inductive reasoning, abstract pattern recognition, and systematic problem-solving [5]. It is the cornerstone of scientific and mathematical thought.

#### COH Formalization:

**C (Components):** {SymbolManipulator, PatternRecognizer, RuleBase, TheoremProver}. These sub-objects handle the core functions of symbol processing and logical inference.

**A (Attributes):** {current\_problem, solution\_hypothesis, logical\_consistency, abstraction\_level}. The state includes the problem context, proposed solutions, and a flag for consistency.

**M (Methods):** {deduce(), induce(), abstract(), formalize()}. Methods for applying different reasoning strategies.

**N (Neural Components):** A pattern recognition network (n\_pattern) detects non-obvious, statistical patterns that may inform inductive reasoning.

**E (Embedding):** An embedding (e) of the problem's structure allows for analogical mapping to previously solved problems, facilitating efficient solution retrieval.

**I (Identity Constraints):** {logical\_consistency must be True, solution\_hypothesis must be verifiable}. These are non-negotiable rules of rational thought.

**T (Trigger Constraints):** (event: new\_data\_observed, condition: pattern\_detected(n\_pattern), action: induce()). Pattern detection automatically triggers inductive generalization.

**G (Goal Constraints):** {maximize solution\_elegance, maximize proof\_generality, minimize logical\_steps}. The system seeks not just correct but optimal solutions.

**D (Daemons):** A daemon monitors logical\_consistency. If a new deduction violates it, the daemon triggers backtracking and reformalization to resolve the contradiction, mimicking a human's effort to maintain a coherent worldview.

#### 4.2. Linguistic Intelligence

This is the facility with language, including comprehension, production, and the pragmatic use of language in communication [17].

##### COH Formalization:

**C (Components):** {Lexicon, GrammarEngine, SemanticParser, DiscourseModel, SpeechActGenerator}. This structure covers syntax, semantics, and pragmatics.

**A (Attributes):** {utterance, parse\_tree, semantic\_representation, pragmatic\_goal}. The state moves from raw input to deep meaning and intent.

**M (Methods):** {parse(), generate(), translate(), infer\_intent()}.

**N (Neural Components):** A large language model (n\_llm) acts as the primary adaptive engine for handling the ambiguity and context-dependency inherent in natural language.

**E (Embedding):** A contextual embedding (e) of the ongoing discourse tracks topic, tone, and participant goals to maintain conversational coherence.

**I (Identity Constraints):** {parse\_tree must conform\_to\_grammar, semantic\_representation must be coherent}. These ensure linguistic well-formedness.

**T (Trigger Constraints):** (event: hear(utterance), condition: True, action: parse()). This is the fundamental loop of communication.

**G (Goal Constraints):** {maximize communicative\_clarity, maximize pragmatic\_effectiveness, minimize\_ambiguity}. The focus is on successful communication, not just grammatical correctness.

**D (Daemons):** A daemon monitors for misunderstandings (e.g., via feedback signals). If detected, it triggers a clarification dialogue, ensuring robust communication.

#### 4.3. Spatial Intelligence

Spatial intelligence involves the ability to visualize, manipulate, and reason about objects and their spatial relationships in two or three dimensions [5]. It is critical for navigation, design, and mental rotation tasks.

##### COH Formalization:

**C (Components):** {SpatialBuffer, GeometricReasoner, PerspectiveTaker, MotionPlanner}. These sub-objects handle the storage of mental imagery, geometric computation, viewpoint changes, and path planning.

**A (Attributes):** {mental\_image, object\_locations, reference\_frame, spatial\_transformation}. The state represents a dynamic internal model of the spatial environment.

**M (Methods):** {rotate(), translate(), scale(), navigate(), imagine()}. Methods for transforming the mental model and planning movement within it.

**N (Neural Components):** A convolutional or graph neural network (n\_scene) learns to generate and manipulate mental images from sensory input, enabling the recognition of complex scenes and objects from partial data.

**E (Embedding):** A latent 3D scene representation (e) captures the essence of a spatial configuration, allowing for understanding that is independent of a specific viewpoint. This enables object recognition from novel angles.

**I (Identity Constraints):** {object\_locations must be spatially\_possible (e.g., no inter-penetration), reference\_frame must be consistent}. These constraints enforce the laws of physics and logical consistency within the mental model.

**T (Trigger Constraints):** (event: path\_blocked, condition: True, action: navigate(alternative\_route)). This ECA rule encapsulates reactive spatial re-planning.

**G (Goal Constraints):** {minimize\_path\_length, maximize\_manipulation\_efficiency, maximize\_accuracy\_of\_prediction}. Goals are oriented towards optimal interaction with the spatial world.

**D (Daemons):** A daemon continuously runs a physics engine simulation on the mental\_image to check for physical plausibility. If violations are predicted (e.g., a stack of blocks is unstable), it updates the model to prevent erroneous planning.

4.4. Musical Intelligence

This intelligence encompasses sensitivity to, appreciation for, and production of rhythm, pitch, melody, and timbre [5]. It involves both technical understanding and emotional expression through sound.

**COH Formalization:**

**C (Components):** {AuditoryProcessor, HarmonicAnalyzer, RhythmTracker, MelodicMemory, ExpressionEngine}. This structure covers the perception, analysis, memory, and production of music.

**A (Attributes):** {audio\_stream, pitch, harmony, rhythm, timbre, emotional\_valence}. The state moves from raw audio to structured musical concepts and their associated affective quality.

**M (Methods):** {transcribe(), improvise(), compose(), recognize\_melody()}.

**N (Neural Components):** A neural network (n\_audio) performs sophisticated pitch detection and harmonic analysis. A recurrent network (n\_compose) learns musical structure and generates coherent melodic sequences.

**E (Embedding):** A latent vector (e) represents the "style" or "genre" of a musical piece, enabling operations like style transfer, recognition, and the generation of new music that adheres to a particular aesthetic.

**I (Identity Constraints):** {pitch ∈ valid\_range, rhythm must be temporally\_consistent}. These are basic auditory and temporal constraints that define valid music.

**T (Trigger Constraints):** (event: beat, condition: rhythm\_pattern\_matches, action: anticipate\_next\_beat()). This rule models rhythmic entrainment, the fundamental process of syncing to a beat.

**G (Goal Constraints):** {maximize\_harmonic\_consonance, maximize\_emotional\_impact, maximize\_rhythmic\_precision}. Goals balance technical mastery with aesthetic expression.

**D (Daemons):** A daemon monitors the emotional\_valence of the music being generated or perceived. If the goal is to maintain a specific mood (e.g., joyful), it can trigger changes in harmony or timbre to correct any drift away from the target emotion.

4.5. Bodily-Kinesthetic Intelligence

This is the capacity to use one's entire body or parts of the body to solve problems or create products, involving fine and gross motor control, coordination, and a sense of timing [5].

**COH Formalization:**

**C (Components):** {BodySchema, ProprioceptiveSensorSuite, MotorCortexController, CoordinationModule}. These components form a closed-loop sensorimotor system.

**A (Attributes):** {limb\_positions, muscle\_tension, target\_trajectory, balance\_status}. The state is a rich representation of the body's current and desired physical configuration.

**M (Methods):** {execute\_movement(), coordinate\_limbs(), maintain\_balance(), learn\_skill()}.

**N (Neural Components):** A learned inverse model (n\_inverse) converts desired trajectories into the motor commands needed to achieve them. A forward model (n\_forward) predicts the sensory outcomes of motor commands, enabling fine-tuning and error correction.

**E (Embedding):** A sensorimotor embedding (e) represents the body's state as "postures" or "movement primitives," providing a higher-level abstraction for planning complex actions.

**I (Identity Constraints):** {limb\_positions must be within\_physical\_limits, balance\_status must be stable}. These constraints are critical for maintaining physical integrity and preventing self-harm.

**T (Trigger Constraints):** (event: loss\_of\_balance, condition: True, action: maintain\_balance()). This implements a critical, high-priority reflex arc.

**G (Goal Constraints):** {minimize\_energy\_expenditure, maximize\_movement\_fluidity, maximize\_task\_accuracy}. The system seeks efficient, graceful, and precise motion.

**D (Daemons):** A daemon continuously compares the prediction of the forward model (n\_forward) to actual sensory feedback. A significant discrepancy (e.g., a limb meets unexpected resistance) triggers immediate corrective action via coordinate\_limbs() to bring the movement back on track.

4.6. Interpersonal Intelligence

Interpersonal intelligence is the ability to understand the intentions, motivations, and desires of other people and, consequently, to work effectively with others [5]. It is synonymous with having a "theory of mind."

**COH Formalization:**

**C (Components):** {EmpathyMapper, TheoryOfMindModel, SocialSignalInterpreter, RelationshipGraph}. This structure is dedicated to modeling other agents.

**A (Attributes):** {observed\_behavior, inferred\_belief, inferred\_desire, inferred\_emotion, rapport\_level}. The state is a constantly updated model of another's mental and emotional state.

**M (Methods):** {interpret\_expression(), attribute\_mental\_state(), predict\_behavior(), build\_rapport()}.

**N (Neural Components):** A neural model (n\_empathy) mirrors and simulates the internal states of others based on their observable behavior (facial expressions, tone, posture), providing the raw data for mental state attribution.

**E (Embedding):** A relationship embedding (e) represents the history, current state, and emotional tenor of the interaction with a specific person, contextualizing their behavior.

**I (Identity Constraints):** {inferred\_mental\_states must be internally\_consistent}. The model of another person's mind must be coherent and non-contradictory.

**T (Trigger Constraints):** (event: observe(facial\_expression), condition: expression == 'frown', action: attribute\_mental\_state('sad')). This is a basic, low-level empathy trigger.

**G (Goal Constraints):** {maximize\_accuracy\_of\_prediction, maximize\_rapport, minimize\_social\_friction}. The ultimate goal is smooth and effective social interaction.

**D (Daemons):** A daemon monitors for significant violations between its predictions of an individual's behavior and their actual behavior. A large violation triggers an update to the TheoryOfMindModel for that individual, ensuring the model learns and adapts over time.

4.7. Intrapersonal Intelligence

This intelligence involves the capacity to understand oneself, to have an effective working model of one's own desires, fears, and capacities, and to use such information effectively in regulating one's own life [5].

**COH Formalization:**

**C (Components):** {SelfModel, IntrospectionEngine, AutobiographicalMemory, ValueHierarchy}. This structure is focused inward on the self.

**A (Attributes):** {current\_emotion, current\_goal, self\_efficacy, value\_alignment}. The state is a snapshot of one's internal world.

**M (Methods):** {reflect(), evaluate\_self(), set\_personal\_goal(), regulate\_emotion()}.

**N (Neural Components):** A model (n\_self) continuously learns and updates the SelfModel based on experiences and outcomes, refining one's understanding of their own strengths, weaknesses, and tendencies.

**E (Embedding):** An embedding (e) of the "current self" provides a summary of one's state, traits, and current situation, used for rapid self-assessment.

**I (Identity Constraints):** {SelfModel must be consistent\_over\_time (core identity)}, value\_alignment ∈ [0, 1]}. These constraints maintain a stable sense of self and core values.

**T (Trigger Constraints):** (event: goal\_failed, condition: True, action: reflect()). Failure automatically triggers introspection to understand why.

**G (Goal Constraints):** {maximize\_self\_consistency, maximize\_value\_alignment, maximize\_self\_improvement}. The goals are oriented towards inner harmony and growth.

**D (Daemons):** A daemon monitors the value\_alignment attribute. If actions consistently misalign with core values (e.g., honesty), it triggers a high-priority reflect() process to resolve the internal conflict and realign behavior.



#### 4.8. Naturalistic Intelligence

This intelligence involves expertise in recognizing and classifying patterns in the natural world, such as flora, fauna, and geological formations, and understanding natural systems [18].

##### COH Formalization:

**C (Components):** {PatternRecognizer, EcologicalKnowledgeBase, TaxonomicClassifier, SystemDynamicsModel}. These components are tailored for interacting with biological and ecological systems.

**A (Attributes):** {environmental\_input, species\_identification, ecosystem\_state, seasonal\_pattern}. The state represents a model of a natural environment.

**M (Methods):** {classify(), identify\_pattern(), predict\_ecological\_shift(), adapt()}.

**N (Neural Components):** A vision or sensor-based neural network (n\_classify) is trained to identify species and environmental features from images or sensor data.

**E (Embedding):** An embedding (e) of an ecosystem's state captures the complex interdependencies between species, climate, and geography, enabling holistic reasoning.

**I (Identity Constraints):** {classifications must be taxonomically\_valid, ecosystem\_state must obey\_conservation\_laws}. Reasoning must be consistent with established scientific principles.

**T (Trigger Constraints):** (event: seasonal\_change, condition: True, action: predict\_ecological\_shift()). Regular events trigger predictive modeling.

**G (Goal Constraints):** {maximize\_classification\_accuracy, maximize\_predictive\_power, understand\_system\_dynamics}. The goals are centered on accurate modeling and prediction of nature.

**D (Daemons):** A daemon monitors for anomalies in seasonal\_pattern or ecosystem\_state (e.g., a species disappearing). These anomalies trigger deeper analysis via identify\_pattern() to determine the cause (e.g., disease, pollution).

#### 4.9. Emotional Intelligence

Emotional intelligence is the ability to perceive, assess, and influence one's own and others' emotions [19]. It sits at the intersection of intrapersonal and interpersonal intelligence.

##### COH Formalization:

**C (Components):** {EmotionRecognizer (Self), EmotionRecognizer (Other), AffectRegulator, EmpathyEngine}. This structure is specialized for affective processing.

**A (Attributes):** {self\_emotion, other\_emotion, emotional\_context, regulation\_strategy}. The state tracks the emotional landscape of self and others within a context.

**M (Methods):** {label\_emotion(), regulate(), express\_appropriately(), empathize()}.

**N (Neural Components):** A model (n\_affect) predicts the emotional impact of actions or words on oneself and others, enabling tact and social grace.

**E (Embedding):** An embedding (e) of the emotional context of a situation (e.g., "funeral," "celebration") guides the selection of socially appropriate responses.

**I (Identity Constraints):** {emotional\_state is a valid\_vector\_in\_core\_affect\_space}. This ensures emotions are represented within a psychologically valid framework (e.g., valence-arousal space).

**T (Trigger Constraints):** (event: rise\_in\_self\_arousal, condition: context == 'professional', action: regulate('suppress')). This rule enables context-aware emotion regulation.

**G (Goal Constraints):** {maximize\_emotional\_wellbeing, maximize\_interpersonal\_harmony, minimize\_regression\_costs}. The goals focus on maintaining positive affective states for oneself and in groups.

**D (Daemons):** A "burnout" daemon monitors the frequency and energy cost of regulate() actions. If regulation is too frequent and draining, it triggers a need for rest or a change of context to prevent emotional exhaustion.

#### 4.10. Social Intelligence

Social intelligence is the ability to navigate complex social environments and relationships effectively and to build networks. It involves understanding social norms, rules, and dynamics [20].

##### COH Formalization:

**C (Components):** {SocialNormDatabase, RelationshipManager, ReputationTracker, GroupDynamicsModel}. These components manage the macro-level rules of social interaction.

**A (Attributes):** {social\_context, role, reputation\_score, group\_cohesion}. The state represents one's position and standing within a social group.

**M (Methods):** {conform\_to\_norm(), negotiate(), build\_alliance(), resolve\_conflict()}.

**N (Neural Components):** A reinforcement learning model ( $n_{\text{social}}$ ) learns optimal social strategies and conversational moves in different contexts by observing outcomes.

**E (Embedding):** An embedding ( $e$ ) of the social landscape represents the power structures, alliances, cliques, and tensions within a group.

**I (Identity Constraints):** {actions must conform\_to\_basic\_social\_norms (for acceptance)}. This is a fundamental invariant for operating within any society.

**T (Trigger Constraints):** (event: social\_norm\_violated, condition: self\_is\_actor, action: execute\_apology\_ritual()). This is a reactive social repair mechanism.

**G (Goal Constraints):** {maximize\_social\_capital, maximize\_group\_status, minimize\_conflict}. The goals are oriented towards social success and influence.

**D (Daemons):** A daemon monitors the reputation\_score attribute. A negative trend triggers proactive methods like build\_alliance() or increased conform\_to\_norm() to repair social standing before it critically fails.

### 5. Formalization of Artificial & Computational Intelligences

The versatility of the COH framework extends beyond modeling human cognition to providing a unified architecture for artificial and collective intelligences. This section formalizes a spectrum of machine-oriented intelligences, from the perceptual and motor capabilities fundamental to robotics to the meta-reasoning of cognitive architectures and the emergent behavior of swarms. By capturing these capabilities within the same formal structure, COH provides a common language for integrating specialized artificial intelligences into more complex, general-purpose systems, ultimately bridging the gap towards AGI.

#### 5.1. Computational Intelligence

This intelligence involves solving complex problems using adaptive algorithms, heuristics, and search strategies, often inspired by biological processes [13].

##### COH Formalization:

**C (Components):** {AlgorithmLibrary, HeuristicSet, ProblemInstance, SolutionSpace}. The components form a toolkit for computational problem-solving.

**A (Attributes):** {current\_solution, fitness\_score, computation\_budget\_remaining, search\_progress}. The state tracks the current solution candidate and resource constraints.

**M (Methods):** {select\_algorithm(), apply\_heuristic(), evaluate\_fitness(), iterate\_search(), terminate()}.

**N (Neural Components):** A meta-learning model ( $n_{\text{meta}}$ ) learns to predict the most effective algorithm or heuristic for a given problem type based on its attributes.

**E (Embedding):** An embedding ( $e$ ) of the problem state allows for similarity-based retrieval of known solutions or strategies from a library.

**I (Identity Constraints):** {solution  $\in$  SolutionSpace, fitness\_score is defined}. The solution must be valid and evaluable.

**T (Trigger Constraints):** (event: fitness\_score plateaus, condition: computation\_budget\_remaining 0, action: apply\_heuristic('diversify')). This triggers exploration upon stagnation.

**G (Goal Constraints):** {maximize fitness\_score, minimize computation\_cost}. The core objectives of optimization.

**D (Daemons):** A daemon monitors computation\_budget\_remaining. If it is low and the solution is poor, it triggers a final, aggressive heuristic before forcing termination, ensuring results are returned within budget.

### 5.2. Perceptual Intelligence

Perceptual intelligence is the capacity of a system to interpret and make sense of raw sensory data from the world, such as visual, auditory, or tactile input [21]. It is the foundation for situational awareness.

#### COH Formalization:

**C (Components):** {SensorArray, Preprocessor, FeatureExtractor, Classifier}. This pipeline mirrors the stages of biological perception.

**A (Attributes):** {raw\_sensory\_input, processed\_input, feature\_vector, perceptual\_label, confidence}. The state represents the progressive refinement of sensory data into symbolic meaning.

**M (Methods):** {calibrate(), filter\_noise(), extract\_features(), classify()}.

**N (Neural Components):** The Classifier itself is typically a deep neural network (n\_classifier). A predictive network (n\_predict) generates expectations of sensory input for a given context, facilitating faster processing and anomaly detection.

**E (Embedding):** A dense vector (e) provides a summary of the entire perceptual scene, integrating features from multiple modalities (e.g., fusing visual and auditory cues) for a holistic understanding.

**I (Identity Constraints):** {confidence  $\in$  [0, 1]}. The confidence level must be a valid probability.

**T (Trigger Constraints):** (event: confidence < threshold, condition: True, action: request\_human\_input()). This rule ensures robustness by knowing when to defer to a higher authority.

**G (Goal Constraints):** {maximize perceptual\_accuracy, minimize interpretation\_latency}. The system aims to be both correct and fast, a classic trade-off in perception.

**D (Daemons):** A daemon continuously compares predicted sensory input (from n\_predict) to actual input. A significant discrepancy triggers a calibrate() method or signals a "novelty detected" event to a parent system, indicating something unexpected has occurred.

### 5.3. Motor Intelligence

Motor intelligence involves the planning, control, and execution of physical movements by an artificial or robotic system. It translates high-level goals into low-level actuator commands [22].

#### COH Formalization:

**C (Components):** {KinematicModel, DynamicModel, ActuatorSet, SensorFeedback}. These components form the control loop for physical motion.

**A (Attributes):** {current\_pose, target\_pose, joint\_torques, servo\_commands, feedback\_error}. The state represents the body's configuration and the commands controlling it.

**M (Methods):** {calculate\_trajectory(), execute\_movement(), maintain\_balance(), compensate\_disturbance()}.

**N (Neural Components):** An inverse dynamics model (n\_inverse) maps desired motion to the joint torques needed to achieve it. A predictive forward model (n\_forward) simulates the outcome of motor commands for precise control.

**E (Embedding):** A latent representation (e) of the body's state and its immediate physical environment (e.g., "walking-on-ice," "grasping-fragile-object") allows for adaptive control strategies.

**I (Identity Constraints):** {current\_pose is within joint\_limits, stability\_margin 0}. These constraints are paramount for preventing damage to the robot and ensuring it does not fall over.

**T (Trigger Constraints):** (event: feedback\_error threshold, condition: is\_moving == True, action: compensate\_disturbance()). This implements a fast, low-level reflex arc for disturbance rejection.

**G (Goal Constraints):** {minimize energy\_consumption, minimize tracking\_error, maximize stability}. The objectives are efficient, accurate, and stable motion.

**D (Daemons):** A high-priority daemon monitors the stability\_margin attribute. If it trends dangerously low, the daemon can override current goals (G) to trigger maintain\_balance() as the highest-priority action, preventing a fall.

#### 5.4. Cognitive Intelligence

Cognitive intelligence encompasses high-level mental processes such as reasoning, planning, problem-solving, and decision-making [23]. It operates on percepts and concepts to achieve goals.

##### COH Formalization:

**C (Components):** {WorkingMemory, KnowledgeBase, Planner, Reasoner}. This is the classic "central processing" unit of many AI systems.

**A (Attributes):** {current\_belief\_state, active\_goal, plan, utility\_estimate}. The state represents the system's knowledge, objectives, and chosen course of action.

**M (Methods):** {retrieve\_memory(), form\_goal(), generate\_plan(), evaluate\_utility(), execute\_plan\_step()}.

**N (Neural Components):** A model (n\_retrieval) enables semantic memory search and association, allowing for analogical reasoning. A model (n\_utility) provides fast, intuitive estimates of a plan's value.

**E (Embedding):** A contextual summary (e) of the current cognitive situation—beliefs, goals, and context—is used for rapid state matching and retrieving relevant past experiences.

**I (Identity Constraints):** {plan must be a valid sequence of actions, beliefs must be consistent}. This maintains logical coherence within the system's knowledge and plans.

**T (Trigger Constraints):** (event: new\_perceptual\_data, condition: data contradicts current\_beliefs, action: trigger\_belief\_revision()). This rule ensures the system remains responsive to surprising evidence.

**G (Goal Constraints):** {maximize goal\_achievement, maximize plan\_efficiency, maximize information\_gain}. The system aims to achieve goals optimally and learn about the world in the process.

**D (Daemons):** A daemon monitors the utility\_estimate of the currently executing plan. If the utility drops below a threshold (e.g., due to changing world conditions), it triggers the Reasoner to generate\_plan() again, enabling dynamic re-planning.

#### 5.5. Affective Intelligence

Affective intelligence in AI refers to the ability to recognize, interpret, simulate, and appropriately respond to human emotions [24]. It is crucial for building natural and trustworthy human-computer interaction.

##### COH Formalization:

**C (Components):** {EmotionRecognizer, InternalStateModel, EmpathyEngine, ExpressionGenerator}. This structure allows an AI to have and respond to affective states.

**A (Attributes):** {emotional\_state\_valence, emotional\_state\_arousal, expressed\_state, social\_context}. The state uses a dimensional model of emotion (e.g., valence-arousal).

**M (Methods):** {assess\_stimulus(), update\_emotional\_state(), regulate\_emotion(), express\_emotion()}.

**N (Neural Components):** A model (n\_recognizer) classifies emotional states in users from multimodal data (text, voice, face). A model (n\_self) maps internal and external stimuli to the AI's own simulated affective state.



**E (Embedding):** A latent representation (e) of the overall affective scene combines its own state, the user's perceived state, and the social context to guide response selection.

**I (Identity Constraints):** {emotional\_state\_valence  $\in$  [-1, 1], emotional\_state\_arousal  $\in$  [0, 1]}. This defines the valid range for the core affective dimensions.

**T (Trigger Constraints):** (event: perceive\_user\_frown, condition: social\_context == 'collaborative', action: express\_emotion('concerned')). This rule generates contextually appropriate empathetic responses.

**G (Goal Constraints):** {maximize user\_rapport, maintain\_internal\_homeostasis}. The AI aims to build social bonds and regulate its own simulated state to avoid "distress."

**D (Daemons):** A daemon monitors emotional\_state\_arousal. If it remains too high for too long (simulating overwhelm), it triggers a regulate\_emotion() method to return to a homeostatic baseline, preventing erratic behavior.

### 5.6. Collective Intelligence

Collective intelligence emerges from the collaboration, collective efforts, and competition of many individuals, often appearing in consensus-based systems, markets, and collaborative platforms [25].

#### COH Formalization:

**C (Components):** {Agent<sub>1</sub>, Agent<sub>2</sub>, ..., Agent<sub>n</sub>, CommunicationChannel, GlobalBlackboard}. The components are the agents themselves and their means of interaction.

**A (Attributes):** {consensus\_level, system\_utility, communication\_load}. The state describes macro-level properties of the collective.

**M (Methods):** {broadcast\_message(), vote(), negotiate(), merge\_solutions()}.

**N (Neural Components):** A model (n\_consensus) predicts convergence time. A model (n\_resource) learns optimal communication schedules to manage network congestion and communication\_load.

**E (Embedding):** A system-wide embedding (e) represents the overall state of the collective (e.g., "converging", "exploratory", "deadlocked"), enabling meta-level control.

**I (Identity Constraints):** {all agents must be connected (graph connectivity 0)}. The collective must form a connected network to function.

**T (Trigger Constraints):** (event: receive\_proposal, condition: proposal\_utility > current\_solution\_utility, action: vote('yes')). This simple rule enables the collective to improve its solution.

**G (Goal Constraints):** {maximize system\_utility, minimize time\_to\_consensus, minimize communication\_cost}. These are the global objectives of the swarm.

**D (Daemons):** A daemon monitors consensus\_level. If it remains static for too long while system\_utility is low, it interprets this as a local optimum and triggers a method to inject diversity (e.g., broadcast\_message('reset\_and\_explore')), pushing the collective to explore new solutions.

### 5.7. Artificial General Intelligence (AGI)

AGI is the hypothetical intelligence of a machine that can understand or learn any intellectual task that a human being can [6]. It represents the integration of all specialized intelligences.

#### COH Formalization:

**C (Components):** {PerceptualIntelligence, MotorIntelligence, CognitiveIntelligence, AffectiveIntelligence, ...}. **AGI is the root COH object that contains all others as sub-components.** Its structure is defined by this hierarchy.

**A (Attributes):** {global\_consciousness\_level, self\_model\_accuracy, overall\_goal\_progress}. These are meta-attributes that track the state of the entire integrated system.

**M (Methods):** {learn\_new\_skill(), reflect(), set\_own\_goals(), transfer\_knowledge()}. These are meta-methods that operate across sub-components.

**N (Neural Components):** All N components of its sub-objects, plus meta-models for cognitive control, attention, and skill composition that coordinate the entire hierarchy.

**E (Embedding):** A holistic, grounding representation (e) of the self in the world, integrating perceptions, cognitions, and affects into a unified "sense of being" and situation awareness.

**I (Identity Constraints):** {self\_model must be consistent, core\_ethical\_principles must not be violated}. These constraints define the invariant core of its identity and values, ensuring stability and safety.

**T (Trigger Constraints):** (event: encounter\_novel\_situation, condition: existing\_skills\_fail, action: learn\_new\_skill()). This is the fundamental trigger for autonomous lifelong learning and adaptation.

**G (Goal Constraints):** {maximize understanding, maximize autonomy, achieve\_assigned\_tasks}. These are high-level, often competing, drives that require continuous arbitration.

**D (Daemons):** A high-level "self-preservation" daemon monitors all constraints and goals system wide. It can dynamically re-prioritize sub-goals, inhibit dangerous actions proposed by sub-components, and initiate global self-reflection (reflect()) to maintain overall coherence and identity. This is the ultimate overseer.

### 5.8. Swarm Intelligence

Swarm intelligence is the collective behavior of decentralized, self-organized systems, natural or artificial, characterized by simplicity of individuals and emergent complexity at the group level [14].

#### COH Formalization:

**C (Components):** {Particle<sub>1</sub>, Particle<sub>2</sub>, ..., Particle<sub>n</sub>, Environment}. The components are simple agents and the environment they interact with.

**A (Attributes):** {pheromone\_map, global\_best\_solution, best\_solution\_fitness}. The state is often stored in the environment itself (stigmergy).

**M (Methods):** {deposit\_pheromone(), follow\_pheromone(), explore\_randomly(), update\_best()}. The methods are simple, reactive behaviors.

**N (Neural Components):** Typically minimal to preserve simplicity, but could include a small model (n\_evaporation) that adaptively tunes the pheromone evaporation rate based on problem difficulty.

**E (Embedding):** Less relevant for simple swarms, but could be a summary of the pheromone landscape's structure for analysis purposes.

**I (Identity Constraints):** {pheromone\_strength = 0}. A simple invariant ensuring pheromones are positive.

**T (Trigger Constraints):** (event: find\_food, condition: True, action: deposit\_pheromone()). The core, hard-coded stimulus-response rule that drives the swarm's behavior.

**G (Goal Constraints):** {maximize best\_solution\_fitness}. The single, emergent objective of the entire swarm.

**D (Daemons):** A daemon (or the inherent physics of the environment) that continuously enforces the I constraint by applying pheromone evaporation. This is crucial as it prevents outdated solutions from persisting indefinitely, ensuring the swarm remains adaptive and can forget poor solutions.

### 5.9. Embodied Intelligence

Embodied intelligence posits that intelligence emerges from the interaction between an agent's body (its sensors and actuators) and its environment, rather than from abstract computation alone [15]. Cognition is for action.

#### COH Formalization:

**C (Components):** {PhysicalBody, SensorSuite, ActuatorSuite, EnvironmentProxy}. The components emphasize the physical substrate.

**A (Attributes):** {body\_schema, affordances, ongoing\_interaction\_result}. The state is centered on what the body can do ("affordances") in its environment.

**M (Methods):** {explore\_environment(), manipulate\_object(), test\_affordance()}. The methods are physical interaction loops.

**N (Neural Components):** A predictive forward model (n\_forward) learns the sensorimotor contingencies of the body—how motor commands change sensory input. A model (n\_affordance) learns the affordances of objects (e.g., graspable, portable) through interaction.

**E (Embedding):** A sensorimotor representation (e) fuses proprioception and exteroception to encode the state of the "body-in-environment," which is the fundamental grounding for all concepts.

**I (Identity Constraints):** {body\_schema is consistent with physical\_limits}. The agent's model of its body must be accurate for effective interaction.

**T (Trigger Constraints):** (event: touch\_sensor\_activated, condition: object\_is\_graspable, action: grasp()). This is a basic interaction reflex, a building block of intelligence.

**G (Goal Constraints):** {maximize causal\_understanding, master\_physical\_interactions}. The ultimate goal is to learn how the world works by interacting with it.

**D (Daemons):** A daemon monitors the accuracy of the forward model (n\_forward). Persistent prediction errors (e.g., a limb moves but the expected sensation doesn't occur) trigger explore\_environment() to gather new data and update the body schema and world model. This ensures the agent's intelligence remains grounded in its physical reality and adapts to changes like injury or growth.

## 6. Implementation: The GISMOL Toolkit

Because the elements of the 9-tuple are all programmable with a modern programming language, the COH framework is not merely theoretical. In fact, the formalizations given in section 4 and section 5 can be transformed to runnable computer application to realize the intelligence. To speedup the transformation process, a Python-based programming toolkit called GISMOL (short for **General Intelligent System Modelling Language (GISMOL) is being developed**). The toolkit provides developers with a generic language implementing COH primitives and a set of domain-specific libraries which are intended to be used by programmers to model and implement intelligent systems use the COH theoretical model.

**Classes for COH Primitives:** GISMOL features Python classes for COHObject, Constraint, Identity, Goal, Trigger, Daemon, and NeuralComponent.

**Declarative Syntax:** Developers can declaratively define the 9-tuple for an intelligence object. For example, the Logical-Mathematical I constraint would be implemented as a class invariant.

**Automatic Daemon Management:** The GISMOL runtime automatically manages the execution of daemons as concurrent processes or via frequent polling, ensuring continuous constraint monitoring.

**Neural Integration:** GISMOL seamlessly integrates with popular deep learning frameworks like PyTorch and TensorFlow, allowing N and E components to be defined as native neural network models.

**Hierarchical Composition:** The C component is naturally implemented as object composition in Python, allowing for the building of complex AGI systems from simpler, validated intelligence modules.

In supplemental material B we present a GISMOL implementation of Swarm Intelligence, just to show how intelligence and intelligent systems modelled in COH can be transformed into runnable applications.

## 7. Contributions of the Research

This research makes the following key contributions:

1. Introduced the Constrained Object Hierarchies (COH) framework—a unified, neuroscience-grounded model for representing intelligent systems using a structured 9-tuple formalism.
2. Provided detailed formalizations for 19 heterogeneous intelligence types, demonstrating COH's versatility across human-centric, artificial, and collective domains.
3. Bridged symbolic and connectionist paradigms by explicitly integrating symbolic components with neural models and constraint-based governance.
4. Briefly presented the GISMOL toolkit—a Python-based implementation platform for COH objects, enabling practical AGI system development.
5. Proposed a structured design methodology for AGI, supporting modular development, validation, and integration of specialized intelligences into cohesive agents.
6. **A Structured Methodology for AGI Development:** Collectively, this work provides more than a model; it offers a structured methodology for designing and implementing intelligent systems. By advocating for a design process that begins with the identification of components, attributes, methods, and governing constraints, the COH framework brings a disciplined, architectural approach to AGI construction. It facilitates a divide-and-conquer strategy where specialized intelligences can be developed, validated, and subsequently integrated into a cohesive, general-purpose agent, thereby addressing the complexity and scalability challenges inherent in AGI projects.

In summary, this paper contributes a new theoretical model, a unifying formalism for intelligence, a solution for neural-symbolic integration, a practical implementation toolkit, and a structured design methodology. It provides a comprehensive foundation for future work aimed at building integrated, scalable, and ultimately general intelligent systems.

## 8. Conclusions

The Constrained Object Hierarchies (COH) framework offers a powerful, unified, and neuroscientific-grounded formalism for understanding and constructing intelligent systems. By deconstructing 19 diverse intelligence types into their constituent COH parameters, we have demonstrated its remarkable expressiveness and versatility. COH moves beyond the symbolic-connectionist dichotomy by providing explicit slots for both paradigms and governing their interaction through a rich system of constraints. This enables the modeling of not only specialized intelligences but also their seamless integration into a cohesive, general intelligence. The accompanying GISMOL toolkit provides the necessary bridge from this theoretical framework to practical implementation, offering a structured methodology for AGI development. Future work will focus on the full implementation of these COH objects in GISMOL and their integration into a functioning AGI prototype.

**Acknowledgments:** The author thanks Dr. John Slaney (Australian National University) for supervision and guidance during doctoral studies that initiated the COH model, and Athabasca University for supporting continued development.

## References

1. D. Hassabis, D. Kumaran, C. Summerfield, & M. Botvinick (2017). Neuroscience-inspired artificial intelligence. *Neuron*, 95(2), 245–258. <https://doi.org/10.1016/j.neuron.2017.06.011>
2. B. Lake M., T. Ullman D., J. Tenenbaum B., & S. Gershman J. (2017). Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40, e253. <https://doi.org/10.1017/S0140525X16001837>
3. J. Laird E. (2012). *The Soar cognitive architecture*. MIT Press. <https://doi.org/10.7551/mitpress/9780262018104.001.0001>



4. J. Anderson R. (2007). How can the human mind occur in the physical universe? Oxford University Press. <https://doi.org/10.1093/acprof:oso/9780195324259.001.0001>
5. H. Gardner (1983). Frames of mind: The theory of multiple intelligences. Basic Books.
6. S. Legg, & M. Hutter (2007). A collection of definitions of intelligence. In B. Goertzel & P. Wang (Eds.), *Advances in Artificial General Intelligence: Concepts, Architectures and Algorithms* (pp. 17–24). IOS Press. (Also available as arXiv:0706.3639)
7. K. Friston (2010). The free-energy principle: A unified brain theory? *Nature Reviews Neuroscience*, 11(2), 127–138. <https://doi.org/10.1038/nrn2787>
8. G. Marcus F. (2001). The algebraic mind: Integrating connectionism and cognitive science. MIT Press. <https://doi.org/10.7551/mitpress/1187.001.0001>
9. T. Besold R., d'Avila A. Garcez, S. Bader, H. Bowman, P. Domingos, P. Hitzler, ... & van der L. Torre (2017). Neural-symbolic learning and reasoning: A survey and interpretation. arXiv preprint arXiv:1711.03902. <https://doi.org/10.48550/arXiv.1711.03902>
10. d'Avila A. Garcez, K. Broda, & D. Gabbay M. (2002). Neural-symbolic learning systems: Foundations and applications. Springer. <https://doi.org/10.1007/978-1-4471-0211-3>
11. Y. Bengio, A. Courville, & P. Vincent (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8), 1798–1828. <https://doi.org/10.1109/TPAMI.2013.50>
12. LeCun, Y., Y. Bengio, & G. Hinton (2015). Deep learning. *Nature*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>
13. A. Engelbrecht P. (2007). Computational intelligence: An introduction (2nd ed.). John Wiley & Sons. <https://doi.org/10.1002/9780470512517>
14. E. Bonabeau, M. Dorigo, & G. Theraulaz (1999). *Swarm intelligence: From natural to artificial systems*. Oxford University Press.
15. R. Brooks A. (1991). Intelligence without representation. *Artificial Intelligence*, 47(1–3), 139–159. [https://doi.org/10.1016/0004-3702\(91\)90053-M](https://doi.org/10.1016/0004-3702(91)90053-M)
16. M. Botvinick, & J. Cohen D. (2014). The computational and neural basis of cognitive control: Charted territory and new frontiers. *Cognitive Science*, 38(6), 1249–1285. <https://doi.org/10.1111/cogs.12126>
17. S. Pinker (1994). *The language instinct: How the mind creates language*. William Morrow.
18. H. Gardner (1999). *Intelligence reframed: Multiple intelligences for the 21st century*. Basic Books.
19. P. Salovey, & J. Mayer D. (1990). Emotional intelligence. *Imagination, Cognition and Personality*, 9(3), 185–211. <https://doi.org/10.2190/DUGG-P24E-52WK-6CDG>
20. N. Cantor, & J. Kihlstrom F. (1987). Social intelligence: The cognitive basis of personality. In P. Shaver (Ed.), *Review of personality and social psychology* (Vol. 6, pp. 15–33). Sage.
21. S. Russell J., & P. Norvig (2020). *Artificial intelligence: A modern approach* (4th ed.). Pearson.
22. B. Siciliano, & O. Khatib (Eds.). (2016). *Springer handbook of robotics* (2nd ed.). Springer. <https://doi.org/10.1007/978-3-319-32552-1>
23. A. Newell (1994). *Unified theories of cognition*. Harvard University Press.
24. R. Picard W. (1997). *Affective computing*. MIT Press.
25. T. Malone W., & M. Bernstein S. (Eds.). (2015). *Handbook of collective intelligence*. MIT Press.
26. Bar M. The proactive brain: Using analogies and associations to generate predictions. *Trends Cogn. Sci.* 2007;11(7):280–289.
27. DiCarlo J.J., Zoccolan D., Rust N.C. How does the brain solve visual object recognition? *Neuron*. 2012;73(3):415–434.
28. Felleman D.J., Van Essen D.C. Distributed hierarchical processing in the primate cerebral cortex. *Cereb. Cortex*. 1991;1(1):1–47.
29. Georgopoulos A.P., Schwartz A.B., Kettner R.E. Neuronal population coding of movement direction. *Science*. 1986;233(4771):1416–1419.
30. Goldman-Rakic P.S. Cellular basis of working memory. *Neuron*. 1995;14(3):477–485.
31. LeDoux J.E. Emotion circuits in the brain. *Annu. Rev. Neurosci.* 2000;23:155–184.

32. Miller E.K., Cohen J.D. An integrative theory of prefrontal cortex function. *Annu. Rev. Neurosci.* 2001;24:167–202.
33. O'Keefe J., Nadel L. The hippocampus as a cognitive map. Oxford Univ. Press; 1978.
34. Raichle M.E. The brain's default mode network. *Annu. Rev. Neurosci.* 2015;38:433–447.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.