Instruction-Based Self-Supervised Online Training of the Perceptual Subsystem of a Cognitive Robotic Architecture

Sarah Schneider SARAH.SCHNEIDER@AIT.AC.AT
Center for Vision, Automation & Control, Austrian Institute of Technology, Vienna 1210, Austria

Evan Krause EVAN.KRAUSE@TUFTS.EDU
Department of Computer Science, Tufts University, Medford, MA 02155 USA

Daniel Soukup DANIEL.SOUKUP@AIT.AC.AT
Center for Vision, Automation & Control, Austrian Institute of Technology, Vienna 1210, Austria

Matthias Scheutz MATTHIAS.SCHEUTZ@TUFTS.EDU
Department of Computer Science, Tufts University, Medford, MA 02155 USA

Abstract

Traditional AI systems often operate under the closed-world assumption, restricting their ability to adapt in dynamic environments. We propose a cognitive architecture (CA) that expands its perceptual capabilities by generating object prototypes from user-provided natural language descriptions. Each prototype is constructed using superellipsoid primitives, enabling structured and interpretable shape representations. The CA employs these prototypes to train a convolutional parametric shape encoder, using rendering parameterizations as automated ground-truth supervision. Once trained, the CA employs the encoder to infer superellipsoid-based representations from real-world object observations. A bidirectional mapping between superellipsoid parameters and natural language terms allows the CA to translate inferred geometric features into human-understandable descriptions. We detail the design of the prototype representations, the synthetically supervised training pipeline, and the language–geometry mapping process. Experimental results demonstrate that the CA enhances its perceptual repertoire through our structured, interpretable object representations.

1. Introduction

Traditional cognitive architectures like ACT-R and Soar made many simplifying assumptions about perceptions and actions that allow modelers to focus on defining the production rules that implement the model behavior. More recently, with advance of cognitive models on robots, perceptual (and also action) modules were augmented in cognitive robotic architectures to be able to take in real-world inputs rather than symbolic abstractions (e.g., see Soar's now extended vision system Boggs (2025), or ACT-R/E Trafton et al. (2013)). While various cognitive architectures now incorporate visual perception via classical computer vision or deep learning models, these approaches are typically static and inflexible (Kotseruba & Tsotsos, 2020). Even with these extensions, cognitive architectures still require modelers to preconfigure or train vision processing modules offline before

This work is licensed under a Creative Commons Attribution International 4.0 License.

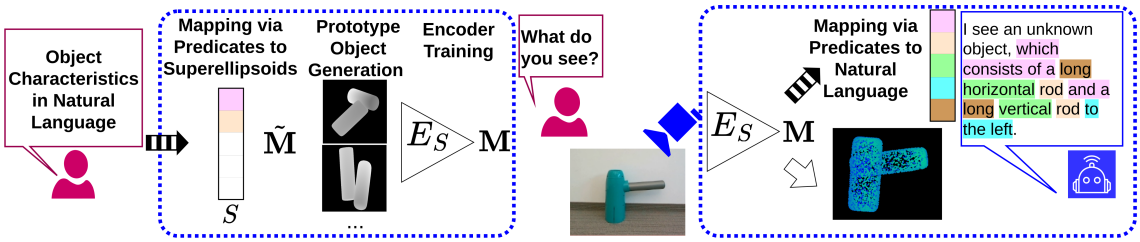


Figure 1. The cognitive architecture (highlighted in blue) collects object characteristics from a user and maps them via object structure predicates S to a superellipsoid-based representation, $\tilde{\mathbf{M}}$. These user-specified characteristics are then used to generate object prototypes and to train a parametric shape encoder model, E_S. Once trained, the encoder can infer the superellipsoid representation $\mathbf{M}$ of unknown objects and generate a corresponding natural language description or schematic visual representation from it.

task execution, so that the robot can recognize task-relevant objects and their pertinent features during operation. In other words, these architecture are still making a closed-world assumption, even if they are intended for showing capabilities such as one-shot object learning Scheutz et al. (2017) or instruction-based task learning that involves novel objects Kirk & Laird (2019). To achieve truly open-world perceptual capabilities, where cognitive agents are able to detect and characterize novel objects (Goel et al., 2024; Boult et al., 2019, 2021), those vision processing modules need to be configured and adapted *online during task performance*. The challenge here is how to make this configuration and adaptation happen *as part of the architecture and during task performance*.

In this paper, we describe a novel extension of the DIARC cognitive robotic architecture (Scheutz et al., 2019) that enables the online configuration and adaptation of its visual object detectors based on given task constraints that enable the detection of large classes of objects. Specifically, given merelogical object characteristics which can be described in natural language, the algorithm synthesizes compact object prototypes that reflect the intended mereological structure. Using these self-generated examples, the architecture autonomously trains a parametric shape encoder capable of inferring geometric object representations from input images. The proposed approach establishes a bidirectional mapping between perception, language, and representation: the inferred representation can be translated back into human-readable descriptions or schematic visualizations. The cognitive architecture is thus able to adapt to new tasks by synthesizing additional object prototypes from novel descriptions, thereby continually expanding its perceptual vocabulary.

2. Background and Motivation

Traditional AI approaches assume a "closed world" where all concepts relevant to the task are known beforehand, and the system model is considered complete. This enables agent designers to craft algorithms based on that predefined knowledge. Consequently, such agents lack the capacity to understand or adapt to information beyond what they were originally programmed to handle. However, the real world is unpredictable and novelties might appear that contradict the previous understanding of the environment. While biological intelligence demonstrates an exceptional ability to robustly adapt to novel inputs, artificial agents remain limited in their ability to replicate this

flexibility (Goel et al., 2024; Boult et al., 2019, 2021; Holder et al., 2025). To effectively operate in open and dynamic environments, a cognitive agent must go beyond merely detecting novelty. It should also be capable of characterizing unfamiliar input, i.e., analyzing and extracting its features and properties, and incorporating this information into its recognition model. This capability enables the agent to manage and respond to similar novel instances in the future (Cruz et al., 2025; Goel et al., 2024).

Depending on its cognitive capabilities, an intelligent system cannot function in isolation and requires external input to generate any form of behavior. However, realistic perception remains challenging in the area of CAs. Notably, almost half of the reviewed architectures in Kotseruba & Tsotsos (2020) do not implement any vision. Some CAs that support visual processing rely on classical computer vision algorithms such as Scale-Invariant Feature Transform (SIFT) (Lowe, 2004), or Local Binary Patterns (LBPs) (Pietikäinen, 2005), sometimes in combination with support vector machines (SVMs) (Cortes & Vapnik, 1995) or mixtures of Gaussians (Reynolds, 2018). Many CAs incorporate pre-trained neural network components for processing perceptual data (Kotseruba & Tsotsos, 2020). For instance, convolutional neural networks (CNNs) trained offline are used for object recognition in the LIDA architecture (Madl et al., 2016). NASA's OnAIR cognitive architecture integrates a deep learning model from the You Only Look Once (YOLO) (Redmon et al., 2016) family as its onboard visual perception module (Zhang et al., 2025). Another model of that family is used for object detction in combination with the OpenCog CA for semantic image retrieval (Potapov et al., 2018).

However, these perception modules are fundamentally static in nature. Classical computer vision methods depend on fixed feature extractors and rule-based classifiers, while even the more advanced deep learning models are typically trained offline and therefore lack the ability to adapt. As a result, such systems are incapable of evolving their perceptual understanding in response to changing environmental conditions.

Open World Object Detection (OWOD) targets recognition in non-stationary environments by flagging unknown instances as unknown without explicit supervision and incrementally learning their categories as labels arrive, without forgetting prior classes (Joseph et al., 2021; Bulzan & Cernazanu-Glavan, 2025). Semi-supervised OWOD (SS-OWOD) reduces reliance on a "human oracle" by using a partially annotated set together with unlabeled data for novel class learning during incremental stages (Mullappilly et al., 2024).

Beyond OWOD and SS-OWOD, online object learning spans several complementary lines of work. These include continual test-time adaptation for detectors, which prescribes when and what to update under distribution shift (Yoo et al., 2024), fully test-time adaptation that performs single-image, pre-prediction updates (Ruan & Tang, 2024), streaming discovery that identifies and consolidates novel objects in video (Kara et al., 2024), and incremental detection methods that stress long-horizon class growth and prototype-based updates (Neuwirth-Trapp et al., 2025; Wang et al., 2025).

While these methods establish the operational loop of detecting and integrating unknowns, they rarely emphasize interpretable representations of object structure. Objects are often modeled as monolithic entities rather than as structured compositions that can be flexibly recombined using existing knowledge.

One avenue toward greater generalizability is compositionality, the principle that complex structures can be formed from constituent parts. Compositionality has been a cornerstone of AI research since its very origins, inspiring developments ranging from neurosymbolic reasoning to modular neural architectures and chain-of-thought reasoning, among others. A key appeal of compositionality lies in its ability to support out-of-distribution generalization while enabling efficient and diverse representations of the world (Elmoznino et al., 2025; Sinha et al., 2024).

Several studies apply the compositional perspective to visual perception by decomposing objects and scenes into semantically meaningful geometric primitives. Gao et al. (2024) and Jiang et al. (2024) propose a hybrid representation that combines superquadrics and 2D Gaussians to learn part-aware 3D scene representations from multi-view images. Alaniz et al. (2023) introduce an optimization based approach to recompose 3D objects into composite superquadrics from multiple views. Ma et al. (2024) reconstruct superquadrics from geometric structures within affordance point clouds. Fedele et al. (2025) propose a method to decompose object point clouds into superquadrics using a neural network followed by Levenberg–Marquardt optimization.

Motivated by these challenges and principles, we develop a system within a CA that can generate object prototypes built from superellipsoids, a flexible class of geometric primitives. From natural language descriptions, the system derives semantic predicates capturing attributes like shape, size, and spatial relations. These predicates are translated into parameter ranges defining the object's configuration, which the cognitive architecture then uses to generate compact synthetic prototypes of the intended structure. The CA utilizes depth views of these synthetic prototypes to train a neural parametric shape encoder, with the corresponding rendering parameters serving as ground truth. Once trained, the encoder can estimate the superellipsoid-based parameterizations from real observed objects. The inferred parameterizations can be mapped back into human-readable descriptions or schematic visualizations of the object, closing the loop between language, perception, and structured representation. This bidirectional mapping enables the CA to adapt to new tasks by synthesizing additional prototypes from novel descriptions, thereby continuously expanding its perceptual vocabulary.

The human-language descriptions can be easily updated to different levels of user expertise, ranging from highly technical, fine-grained terminology to more common everyday language, or even other languages. In addition, schematic visualizations of the generated objects help accommodate users who have difficulty understanding human langauge.

The synthetic objects generated by the system come with automatically derived ground-truth parameters, eliminating the need for manual annotation and enabling synthetic supervision: the cognitive architecture can conduct the training process by itself directly on these data. Because the model outputs superellipsoid-based object representations, their reliability, and the reliability of the associated descriptions, can be quantitatively assessed by comparing the 3D reconstruction with the observed object and effectively communicated to the user.

The system is modular, allowing encoder models to be easily deactivated or replaced within the cognitive architecture when they are no longer needed. Once an object representation is computed, it can be reused for various downstream tasks.

Our contribution can be summarized as: (i) We propose a method for designing synthetic composite objects that serve as fundamental object prototypes. (ii) We formalize a compact and struc-

tured parameterization scheme for representing these objects based on their superellipsoidal geometry. (iii) We leverage this representation to train a convolutional neural network (CNN) on depth-based renderings, using the corresponding object parameters as ground truth for supervision. (iv) We introduce a bidirectional mapping between superellipsoid parameters and human-understandable descriptions, mediated through semantic predicates. (v) We integrate this representation pipeline into a modular cognitive architecture. (vi) We demonstrate the feasibility of our approach through implementation within a CA framework.

3. Superellipsoid-Based Object Prototypes as Compact Object Abstractions

3.1 Superellipsoids as Object components

We represent 3D objects as assemblies of superellipsoids, a flexible, parametric family of shapes derived from superquadrics (Barr, 1981). Superellipsoids enable compact descriptions of diverse geometric forms such as spheres, cylinders, disks, and cuboids, while supporting analytic control over shape and scale parameters.

The surface of a superellipsoid is implicitly defined by:

$$f(\mathbf{x};\mathbf{p},\mathbf{a},\boldsymbol{\epsilon}) = \left(\left(\frac{x-x_0}{a_x}\right)^{\frac{2}{\epsilon_2}} + \left(\frac{y-y_0}{a_y}\right)^{\frac{2}{\epsilon_2}}\right)^{\frac{\epsilon_2}{\epsilon_1}} + \left(\frac{z-z_0}{a_z}\right)^{\frac{2}{\epsilon_1}} = 1, \tag{1}$$

where $\mathbf{x} = (x, y, z) \in \mathbb{R}^3$ is a point in space, $\mathbf{p} = (x_0, y_0, z_0) \in \mathbb{R}^3$ denotes the center of the superellipsoid, $\mathbf{a} = (a_x, a_y, a_z) \in \mathbb{R}^3_+$ are the axis-aligned scaling factors, and $\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2) \in \mathbb{R}^2_+$ are curvature exponents.

The parameter ϵ_1 governs vertical curvature along the z-axis, while ϵ_2 controls curvature in the xy-plane. Varying these parameters allows interpolation between common geometric primitives, including spheres ($\epsilon_1 = \epsilon_2 = 1$), cubes ($\epsilon_1, \epsilon_2 \to 0$), and cylinders or disks (mixed values).

To support arbitrary orientations, we extend the formulation with two Euler angles: azimuth $\phi \in [-\frac{\pi}{2}, \frac{\pi}{2}]$, representing rotation about the vertical z-axis, and elevation $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$, representing rotation about the horizontal x-axis. We denote the combined rotation vector as $\boldsymbol{\psi} = (\phi, \theta)$.

A point $\mathbf{x}$ is transformed into the component's local frame via:

$$\mathbf{x}' = \mathbf{R}_{\text{euler}}^{\top}(\mathbf{x} - \mathbf{p}), \tag{2}$$

where $\mathbf{R}_{\text{euler}} = \mathbf{R}_z(\phi)\mathbf{R}_x(\theta) \in SO(3)$ is the composite rotation matrix constructed from the azimuth and elevation angles.

Each component is thus parameterized by its center $\mathbf{p}$, axis scales $\mathbf{a}$, shape exponents $\boldsymbol{\epsilon}$, and orientation vector $\boldsymbol{\psi}$, enabling compact and expressive descriptions of 3D parts.

3.2 Object Composition

An object O is defined as a collection of N superellipsoidal components:

$$O = \{C_1, C_2, \ldots, C_N\}, \tag{3}$$

where each component corresponds to the region:

$$C_i = \{\mathbf{x} \in \mathbb{R}^3 \mid f(\mathbf{x}; \mathbf{p}_i, \mathbf{a}_i, \boldsymbol{\epsilon}_i, \boldsymbol{\psi}_i) \leq 1\}. \tag{4}$$

To ensure geometric connectivity, we designate the first component C_1 as the root and require all others to be positioned within a fixed threshold $\delta > 0$ from it:

$$\|\mathbf{p}_i - \mathbf{p}_1\| \leq \delta, \quad \forall i > 1. \tag{5}$$

The final object is then constructed as the union of all components:

$$O = \bigcup_{i=1}^{N} C_i. \tag{6}$$

To standardize alignment, each object is uniformly scaled and translated to fit inside the reference unit cube $[-1, 1]^3$ centered at the origin, providing a consistent coordinate frame.

3.3 Structured Object Encodings for Parametric Shape Encoder Training via Synthetic Supervision

To encode the object structure in a learnable form, we construct a semantic matrix $\mathbf{M} \in \mathbb{R}^{10\times N}$ that captures the parameters of all superellipsoidal components in a spatially consistent format. Each component C_i is parameterized by its center position $\mathbf{p}_i = (x_{0_i}, y_{0_i}, z_{0_i}) \in \mathbb{R}^3$, shape exponents $\boldsymbol{\epsilon}_i = (\epsilon_{1_i}, \epsilon_{2_i}) \in \mathbb{R}^2$, scale parameters $\mathbf{a}_i = (a_{x_i}, a_{y_i}, a_{z_i}) \in \mathbb{R}^3$, and rotation angles $\boldsymbol{\psi}_i = (\phi_i, \theta_i) \in \mathbb{R}^2$, representing azimuth and elevation. For consistent representation, we define a permutation $\sigma : 1, \ldots, N \to 1, \ldots, N$ that orders components lexicographically based on their spatial centers. Specifically, a component $\mathbf{p}_i$ is considered to precede $\mathbf{p}_j$ (denoted $\mathbf{p}_i \succ \mathbf{p}_j$) if it lies higher along the z-axis, or, if tied, closer along the y-axis, and subsequently along the x-axis:

$$\mathbf{p}_i \succ \mathbf{p}_j \iff z_{0_i} > z_{0_j} \ \vee \ (z_{0_i} = z_{0_j} \wedge y_{0_i} < y_{0_j}) \ \vee \ (z_{0_i} = z_{0_j} \wedge y_{0_i} = y_{0_j} \wedge x_{0_i} < x_{0_j}). \tag{7}$$

The semantic matrix $\mathbf{M} \in \mathbb{R}^{10\times N}$ is then constructed as:

$$\mathbf{M} = \begin{pmatrix} \mathbf{p}_{\sigma(1)} & \cdots & \mathbf{p}_{\sigma(N)} \\ \boldsymbol{\epsilon}_{\sigma(1)} & \cdots & \boldsymbol{\epsilon}_{\sigma(N)} \\ \mathbf{a}_{\sigma(1)} & \cdots & \mathbf{a}_{\sigma(N)} \\ \boldsymbol{\psi}_{\sigma(1)} & \cdots & \boldsymbol{\psi}_{\sigma(N)} \end{pmatrix}, \tag{8}$$

where each column corresponds to a single superellipsoid component, ordered according to the spatial permutation σ. This structured representation guides the training of a convolutional neural network (CNN), the parametric shape encoder E, enabling it to predict a superellipsoid-based parmetrization from an object observation.

3.4 Predicate-Grounded Bidirectional Mapping Between Superellipsoid Parameters and Natural language

The object representation in the semantic matrix $\mathbf{M}$ offers a compact, structured description of an object's geometry. However, the encoded parameters are not inherently intuitive for human interpretation. To facilitate a more natural understanding of superellipsoid representations, we translate these parameters into semantic predicates that capture the object's geometric characteristics. We first define a COMPOSITION predicate, $HasComp(O, C_i)$, indicating whether an object O includes a specific component C_i with $j \in 1, \ldots, N$. This predicate reflects the number of components an object consists of. Notably, $HasComp(O, C_1)$ is always true, since every object has at least one component. For each identified component C_i, we define SHAPE predicates like $Disk(C_i)$ or $Rod(C_i)$, based on the component's principal axes $\mathbf{a}_i$ and curvature $\boldsymbol{\epsilon}_i$. If a component does not fit any specific shape predicate, we assign a generic $Part(C_i)$ predicate as a placeholder for undefined shapes. For components that are not the first identified component, we categorize their parameters encoding position into POSITION predicates such as $OnTop(C_i, C_1)$ or $ToTheLeft(C_i, C_1)$, relative to the first component C_1. Similarly, we use the SIZE predicates to represent the relative differences in size between the components, such as $Smaller(C_i, C_1)$ or $Larger(C_i, C_1)$, based on the comparison of their volumes derived from the primary axes scales $\mathbf{a}_1$ and $\mathbf{a}_i$.

For instance, consider an object composed of two components: a flat, circular base and a tall, cylindrical extension. The object's structure can be represented by the combination of the predicate $Disk(C_1)$, which captures the base's curvature and axis proportions, and $Rod(C_2)$ for the second component, reflecting its elongated cylindrical shape. Their relative vertical arrangement is expressed through the spatial predicate $OnTop(C_2, C_1)$. A full list of our predicates grounded in superellipsoid parameters is given in Table 1. These predicates are readily adaptable to varying levels of user expertise. Specifically, we use $Rod(C_i)$ as a more familiar alternative to the $Cylinder(C_i)$ predicate, and $Block(C_i)$ as a more accessible equivalent to the $Cuboid(C_i)$ predicate.

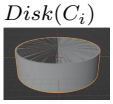
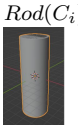
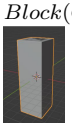
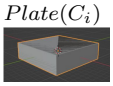
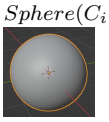
4. Cognitive Architecture Integration Framework

We integrate our method into the Distributed Interactive Affect Reflection Cognition (DIARC) architecture (Scheutz et al., 2019), a cognitive robotic architecture composed of specialized components that communicate via messages expressed in logical form. Fig. 2 illustrates the DIARC configuration used in this work. The language understanding components derive logical expressions interpretable by other modules of the system. The dialogue manager coordinates communication with the user and submits goals to the goal manager, which can issue action scripts or forward search requests to the vision component. A knowledge base stores what the system knows about the environment. Within the vision component, we embed prototype object generation, encoder training, and the mapping from superellipsoid parameters to natural language predicates.

4.1 Autonomous Synthetic Object Generation and Encoder Training from User Instructions

To enable autonomous training data generation, the system accepts high-level instructions that specify an *object structure* S. We define S as the geometric description of an object's composition using

Table 1. Criteria for mapping superellipsoid parameters to semantic predicates. Here, O represents the object being described, composed of components C_i.

Predicate Type	Predicate	Predicate Constraints
COMPOSITION	$HasComp(O, C_1)$	Always true ($\top$)
	$HasComp(O, C_i)$	$\sum m_i > t_{ex}$ where $i \in \{2, \ldots, N\}$
SHAPE	$Disk(C_i)$ 	$HasComp(O, C_i) \wedge \epsilon_{1_i} \approx 0 \wedge \epsilon_{2_i} \approx 1 \wedge a_{x_i} \geq 2a_{z_i} \wedge a_{x_i} \approx a_{y_i}$
	$Rod(C_i) \mid Cylinder(C_i)$ 	$HasComp(O, C_i) \wedge \epsilon_{1_i} \approx 0 \wedge \epsilon_{2_i} \approx 1 \wedge a_{z_i} \geq 2a_{x_i} \wedge a_{x_i} \approx a_{y_i}$
	$Block(C_i) \mid Cuboid(C_i)$ 	$HasComp(O, C_i) \wedge \epsilon_{1_i} \approx 0 \wedge \epsilon_{2_i} \approx 0$
	$Plate(C_i)$ 	$HasComp(O, C_i) \wedge \epsilon_{1_i} \approx 0 \wedge \epsilon_{2_i} \approx 0 \wedge a_{x_i} \geq 2a_{z_i} \wedge a_{x_i} \approx a_{y_i}$
	$Sphere(C_i)$ 	$HasComp(O, C_i) \wedge \epsilon_{1_i} \approx 1 \wedge \epsilon_{2_i} \approx 1 \wedge a_{x_i} \approx a_{y_i} \approx a_{z_i}$
	$Capsule(C_i)$	$HasComp(O, C_i) \wedge \epsilon_{1_i} \approx 1 \wedge \epsilon_{2_i} \approx 1 \wedge \wedge a_{z_i} \geq 2a_{x_i} \wedge a_{x_i} \approx a_{y_i}$
	$Part(C_i)$	$HasComp(O, C_i) \wedge \neg(Disk(C_i) \vee Rod(C_i) \vee Block(C_i) \vee Plate(C_i) \vee Sphere(C_i) \vee Capsule(C_i))$ where $i \in \{1, \ldots, N\}$
ROTATION	$Unrotated(C_i,)$	$HasComp(O, C_i) \wedge (Rod(C_i) \vee Block(C_i) \vee Disk(C_i) \vee Plate(C_i)) \wedge \phi_i \approx 0 \wedge \theta_i \approx 0$
	$Horizontal(C_i,)$	$HasComp(O, C_i) \wedge (Rod(C_i) \vee Block(C_i)) \wedge \phi_i \approx 90$
	$Vertical(C_i,)$	$HasComp(O, C_i) \wedge (Rod(C_i) \vee Block(C_i)) \wedge \phi_i \approx 0$ where $i \in \{1, \ldots, N\}$
POSITION	$ToTheLeft(C_i, C_1)$	$HasComp(O, C_i) \wedge y_{0_i} \lesssim y_{0_1}$
	$ToTheRight(C_i, C_1)$	$HasComp(O, C_i) \wedge y_{0_i} \gtrsim y_{0_1}$
	$AtTheBottom(C_i, C_1)$	$HasComp(O, C_i) \wedge z_{0_i} \lesssim z_{0_1}$
	$OnTop(C_i, C_1)$	$HasComp(O, C_i) \wedge z_{0_i} \gtrsim z_{0_1}$ where $i \in \{2, \ldots, N\}$
ECCENTRICITY	$Long(C_i)$	$HasComp(O, C_i) \wedge (Rod(C_i) \vee Block(C_i)) \wedge a_{z_i} \geq 2.5a_{x_i}$ where $i \in \{1, \ldots, N\}$
SIZE	$Smaller(C_i, C_1)$	$HasComp(O, C_i) \wedge a_{x_i}a_{y_i}a_{z_i} \lesssim a_{x_1}a_{y_1}a_{z_1}$
	$Larger(C_i, C_1)$	$HasComp(O, C_i) \wedge a_{x_i}a_{y_i}a_{z_i} \gtrsim a_{x_1}a_{y_1}a_{z_1}$
	$SimilarSized(C_i, C_1)$	$HasComp(O, C_i) \wedge a_{x_i}a_{y_i}a_{z_i} \approx a_{x_1}a_{y_1}a_{z_1}$ where $i \in \{2, \ldots, N\}$

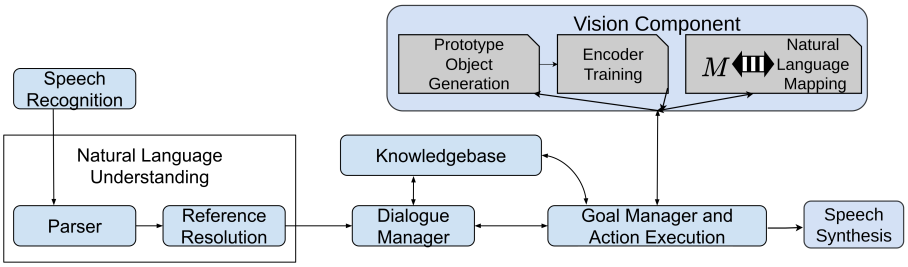


Figure 2. Our object generation, encoder training and natural language mapping are integrated into the vision component of the illustrated DIARC configuration.

the semantic predicates defined in Sec. 3.4. Users provide these specifications through a template-based dialogue interface. From this input, the system extracts semantic predicates (see Tab. 1) and maps them to the corresponding rendering parameters for generating synthetic objects.

When certain attributes of S are left unspecified, the system samples them from predefined uniform distributions. For example, if the SHAPE predicate is omitted for a component $C_{i,S}$, the superellipsoid curvature exponents $\boldsymbol{\epsilon}_{i,S}$ and scale parameters $\mathbf{a}_{i,S}$ are drawn from $\mathcal{U}(0,1)$. If the spatial POSITION of C_i is unspecified, the center coordinates $\mathbf{p}_{i,S}$ are sampled independently along each axis from $\mathcal{U}(-1,1)$. If no SIZE predicate is given, $\mathbf{a}_{i,S}$ is again sampled from $\mathcal{U}(0,1)$ without constraints. However, when a size predicate specifies a relative size with respect to the first component $C_{1,S}$, sampling is restricted so that the resulting volumes satisfy the stated relation.

Rotation constraints are handled similarly. If the user specifies $Unrotated(C_{i,S})$, the azimuth and elevation angles $\boldsymbol{\psi}_{i,S}$ are fixed to zero ($\phi_{i,S} = 0$, $\theta_{i,S} = 0$), otherwise they are drawn from $\mathcal{U}(-\frac{\pi}{2}, \frac{\pi}{2})$. Based on the combination of user-specified constraints and sampled parameters, the system generates a dataset of synthetic object instances that conform to S.

The system supports varying levels of specification, from fully defined structures to partially constrained or entirely unconstrained objects. Table 2 illustrates four example categories. Object structure category S_{I} contains generic three-component objects with no additional constraints. Object structure S_{II} specifies the first component as an unrotated disk and the second as an unrotated rod, both without position constraints. Object structure S_{III} consists of two rod components with unconstrained positions and orientations. Object structure S_{IV} features a block base, a sphere positioned to its right, and a disk placed on top, all with random rotations.

The three-dimensional prototype objects are rendered in Blender (Blender Online Community, 2018), with rendering parameters stored alongside the corresponding object mesh data. For each object, a depth image $\mathbf{I} \in \mathbb{R}^{250\times250}$ is generated using a simple orthogonal projection onto a plane aligned with the x- and y-axes. The blender depth sensor is positioned along the z-axis at a distance equal to twice the side length of the reference cube the prototype object is places into. Depth images capture the geometric structure of the objects while remaining unaffected by appearance attributes such as color, texture, or lighting.

The depth images are processed by a small convolutional network, the parametric shape encoder E_S. The encoder is trained to predict the semantic matrix $\mathbf{M} = E_S(\mathbf{I})$. As supervision, we use

the ground-truth matrix $\tilde{\mathbf{M}} \in \mathbb{R}^{10\times N}$, constructed from the rendering parameters (see Sec. 3.3). Training minimizes the mean squared error (MSE) between $\mathbf{M}$ and $\tilde{\mathbf{M}}$, thereby aligning the encoder output with the superellipsoid object parametrization.

In our current integration, the cognitive architecture (CA) generates 50 prototype objects for each specified object structure, allocating 40 samples for training and 10 for validation. The encoder architecture consists of three convolutional layers followed by two fully connected layers. Each convolutional block includes a convolutional layer with a kernel size of 3×3 and unit padding, followed by a LeakyReLU activation and a 2×2 max-pooling operation to progressively reduce spatial dimensionality while capturing hierarchical spatial features.

The flattened output from the final convolutional layer is passed through a fully connected layer with LeakyReLU activation, followed by a second linear layer that projects the features into the same embedding space as $\tilde{\mathbf{M}}$. Optimization is performed using the Adam optimizer (Kingma & Ba, 2014) with a learning rate of 10^{-6} and a batch size of 16. Training runs for 2000 epochs, after which the user is notified upon completion.

4.2 From Superellipsoid Parameters to Multimodal Object Descriptions

After training the parametric shape encoder E_S and notifying the user, the system can be queried with the prompt *"What do you see?"* to produce descriptions of the observed objects.

4.2.1 Generating Human-Readable Descriptions from Superellipsoid Representations

The system acquires visual input via a depth sensor, using the RGB stream for object segmentation with the Fast Segment Anything Model (Zhao et al., 2023), which produces real-time two-dimensional masks for each frame. These masks are then fused with the sensor's depth data to construct three-dimensional object point clouds. A depth view of the object is produced by replicating the same process used for training data generation. Specifically, the object's point cloud is translated and uniformly scaled to fit within a reference unit cube. Depth views are then obtained through orthogonal projection onto a plane aligned with the x and y axes.

The CA employs the trained encoder model E_S to infer a superellipsoid-based representation of the observed object in the depth image $\mathbf{I}$. For interpretability, the information captured in $\mathbf{M} = E_S(\mathbf{I})$ is mapped back to our predefined set of semantic predicates. These predicates are subsequently used to populate predefined text templates. Template selection is determined by the COMPOSITION predicates, while additional attributes, such as SHAPE and POSITION, are used to fill the corresponding entries. An object with a single component is described by its eccentricity, rotation, and shape. If two components are present, the second is added with its eccentricity, rotation, shape, relative size and position. Descriptions of objects with three or more components are constructed analogously, by successively including each additional component together with its determined attributes. The same mapping procedure can be easily adapted to other languages. Note that in our current implementation, the system uses the most recently added parametric shape encoder if a user prompts it to describe an object.

Table 2. Examples of four different prototype object structures ($S_{\mathrm{I}}, S_{\mathrm{II}}, S_{\mathrm{III}}, S_{\mathrm{IV}}$). Objects from each category are constructed based on given user input descriptions. Notably, if object properties (i.e. predicates such as SHAPE or POSITION) are not specified, the system generates the objects based on randomly sampling parameters from uniform distributions $\mathcal{U}$. Highlighted cells indicate the presence of specific predicates defined for the object strcuture.

Object Structure	S_{I}	S_{II}	S_{III}	S_{IV}
User Input	Object structure characteristics are acquired via natural language dialogue			
COMPOSITION	$HasComp(O_{\mathrm{I}}, C_{1,\mathrm{I}})$ $HasComp(O_{\mathrm{I}}, C_{2,\mathrm{I}})$ $HasComp(O_{\mathrm{I}}, C_{3,\mathrm{I}})$	$HasComp(O_{\mathrm{II}}, C_{1,\mathrm{II}})$ $HasComp(O_{\mathrm{II}}, C_{2,\mathrm{II}})$	$HasComp(O_{\mathrm{III}}, C_{1,\mathrm{III}})$ $HasComp(O_{\mathrm{III}}, C_{2,\mathrm{III}})$	$HasComp(O_{\mathrm{IV}}, C_{1,\mathrm{IV}})$ $HasComp(O_{\mathrm{IV}}, C_{2,\mathrm{IV}})$ $HasComp(O_{\mathrm{IV}}, C_{3,\mathrm{IV}})$
SHAPE	$a_{x_{i,\mathrm{I}}}, a_{y_{i,\mathrm{I}}}, a_{z_{i,\mathrm{I}}} \sim \mathcal{U}$ $\epsilon_{1_{i,\mathrm{I}}}, \epsilon_{2_{i,\mathrm{I}}} \sim \mathcal{U}$	$Disk(C_{1,\mathrm{II}})$ $Rod(C_{2,\mathrm{II}}) \mid Cylinder(C_{2,\mathrm{II}})$	$Rod(C_{1,\mathrm{III}}) \mid Cylinder(C_{1,\mathrm{III}})$ $Rod(C_{2,\mathrm{III}}) \mid Cylinder(C_{2,\mathrm{III}})$	$Block(C_{1,\mathrm{IV}}) \mid Cuboid(C_{1,\mathrm{IV}})$ $Sphere(C_{2,\mathrm{IV}})$ $Disk(C_{3,D})$
ROTATION	$\phi_{i,\mathsf{A}}, \theta_{i,\mathsf{A}} \sim \mathcal{U}$	$Unrotated(C_{1,B})$ $Unrotated(C_{2,\mathrm{II}})$	$\phi_{i,\mathsf{A}}, \theta_{i,\mathsf{A}} \sim \mathcal{U}$	$\phi_{i,\mathsf{A}}, \theta_{i,\mathsf{A}} \sim \mathcal{U}$
POSITION	$x_{0_{i,\mathrm{I}}}, y_{0_{i,\mathrm{I}}}, z_{0_{i,\mathrm{I}}} \sim \mathcal{U}$	$x_{0_{i,\mathrm{II}}}, y_{0_{i,\mathrm{II}}}, z_{0_{i,\mathrm{II}}} \sim \mathcal{U}$	$x_{0_{i,\mathrm{III}}}, y_{0_{i,\mathrm{III}}}, z_{0_{i,\mathrm{III}}} \sim \mathcal{U}$	$ToTheRight(C_{2,\mathrm{IV}}, C_{1,\mathrm{IV}})$ $OnTop(C_{3,\mathrm{IV}}, C_{1,\mathrm{IV}})$
SIZE		$Larger(C_{2,\mathrm{II}}, C_{1,\mathrm{II}})$		
Object				
Depth View				

4.2.2 Rendering Superellipsoids as Schematic Visual Representations

The semantic matrix $\mathbf{M}$ comprises the full set of estimated parameters describing an object's superellipsoid components. Leveraging these parameters, a 3D model of the object can be generated based on superellipsoid geometry (refer to Eq. 1). For simpler visualization, the three-dimensional shape is projected onto the $x - y$ plane, resulting in a two-dimensional representation. Such visual abstractions serve as an alternative approach to communicate object features, which may be particularly helpful for individuals who find interpreting verbal or textual descriptions challenging.

4.3 Estimating Description Confidence

The structured object representation provides a foundation for assessing the reliability of the computed parametrization. By comparing it with the actual object it models, one can quantify the system's confidence in its computed description.

The parameters in the semantic matrix $\mathbf{M}$ can be used to reconstruct a three-dimensional object based on superellipsoids. This is achieved by generating a point cloud $C_{\mathbf{M}} = \{\mathbf{u}_k = (x_k, y_k, z_k) \mid k = 1, \ldots, K\}$ by sampling K points from the surface of the superellipsoids defined by $\mathbf{M}$. Each point $\mathbf{u}_k$ lies in $\mathbb{R}^3$.

Similarly, the point cloud $C_{\mathbf{I}} = \{\mathbf{v}_l = (x_l, y_l, z_l) \mid l = 1, \ldots, L\}$ represents the L points obtained from the object depicted in the depth image $\mathbf{I}$.

The geometric discrepancy between these two point clouds is quantified using the Chamfer distance (Dubuisson & Jain, 1994):

$$d(C_{\mathbf{M}}, C_{\mathbf{I}}) = \frac{1}{2}\left(\frac{1}{K}\sum_{\mathbf{u}\in C_{\mathbf{M}}} \min_{\mathbf{v}\in C_{\mathbf{I}}} \|\mathbf{u}-\mathbf{v}\|^2 + \frac{1}{L}\sum_{\mathbf{v}\in C_{\mathbf{I}}} \min_{\mathbf{u}\in C_{\mathbf{M}}} \|\mathbf{v}-\mathbf{u}\|^2\right)$$

A smaller Chamfer distance $d(C_{\mathbf{M}}, C_{\mathbf{I}})$ indicates that the parameters in $\mathbf{M}$ closely match the true object geometry, leading to a more reliable description. This uncertainty metric is applicable to any object, whether known or novel, synthetic or real. Its deterministic formulation ensures a transparent and reproducible confidence measure. For consistency, the object point cloud $C_{\mathbf{I}}$ is scaled and translated to fit within the reference cube, i.e., the coordinate frame in which the superellipsoid representation resides. This normalization constrains the Chamfer distance to a predictable range. Therefore, the system can flag a computed description with low-confidence statement whenever the distance exceeds a predefined threshold t_{CM}, selected based on the dimensions of the reference cube. In such cases, the system explicitly reports: "I have low confidence in my generated description." in addition to the generated description.

5. Experimental Demonstration

We demonstrate the ability of our system to adapt to different configurations of an unknown object. Specifically, we consider a tennis ball launcher machine, which can be physically configured in two ways: either as a single unit, denoted as object O_A, or as a two-part assembly, denoted as object O_B, formed by manually attaching or detaching an additional component. Our DIARC configuration

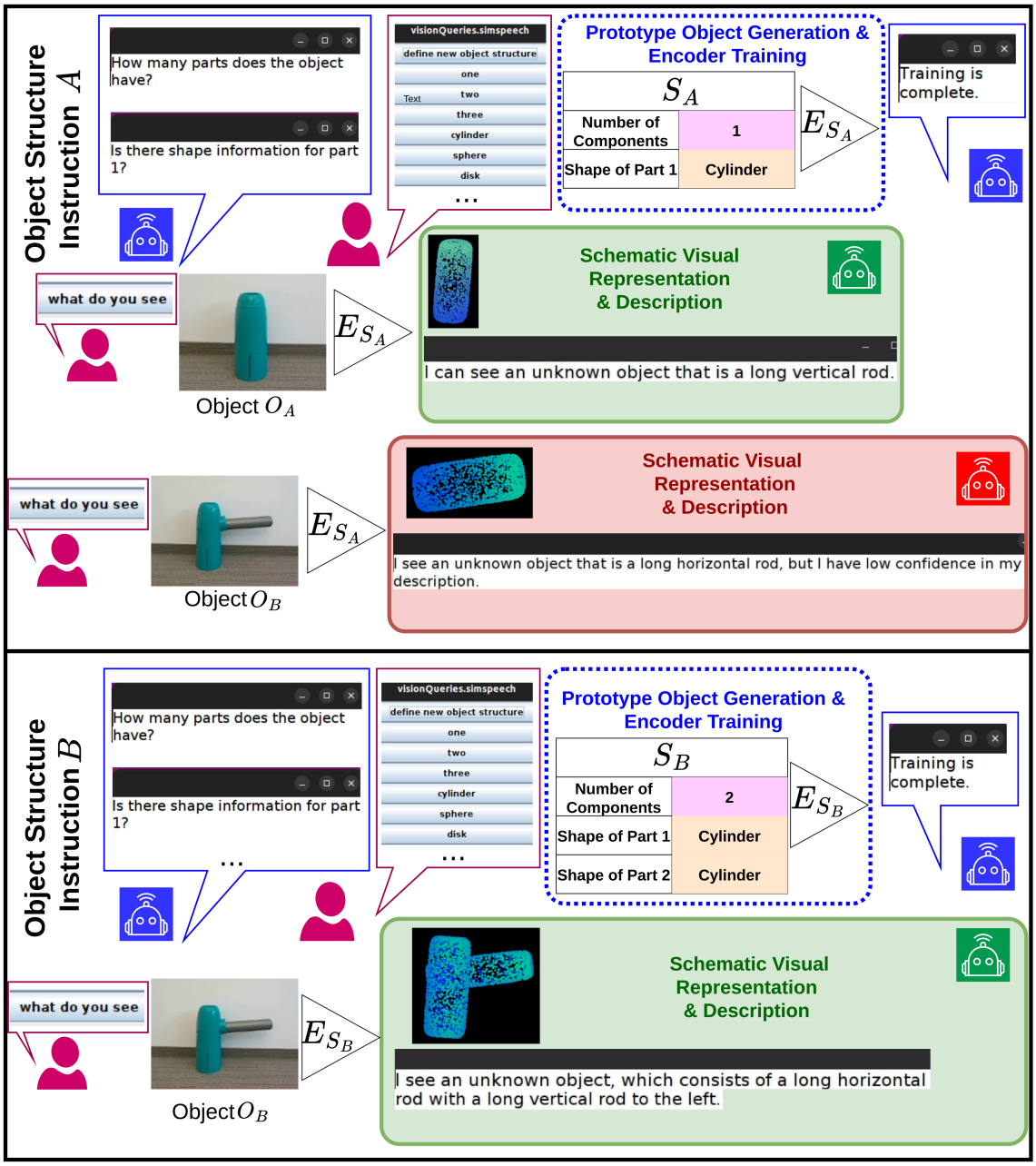


Figure 3. After automated training on a single-cylinder object structure S_A, the system generates an accurate description for object O_A using its trained parametric encoder model E_{S_A}. When presented with the two-cylindrical object O_B, the system's inferred representation can not generate an adequate description and it communicates its uncertainty. After training on the two-cylinder object structure S_B, the system uses the newly trained model E_{S_B} to generate an accurate description of O_B.

runs on a laptop connected to a Intel RealSense depth sensor. User interaction with the system is mediated through a graphical user interface (GUI), where the user can define object structures by answering system queries via predefined button selections and prompt the system as needed.

The process begins when the user selects "`define new object structure`". The system then prompts the user to specify the number of object parts and their corresponding shapes.

In a first dialogue-based instruction, the user defines an object structure S_A consisting of a single cylindrical part. Based on this specification, the system generates prototype objects, i.e. varieties of one-cylinder objects with randomly sampled orientations. It then trains the encoder on these prototype objects and notifies the user upon completion. Once training is finished, the user presents object O_A in front of the depth sensor. When queried with "`What do you see?`", the system uses the trained model E_{S_A} to generate both a visual representation and a natural-language description:

> "`I see an unknown object that is a long vertical rod.`"

Next, the user places object O_B, the two-part version of the object, in front of the sensor and again prompts the system with "`What do you see?`". The system uses the available parametric shape encoder E_{S_A}, which was trained only on the single-part structure S_A, to process the object and responds:

> "`I see an unknown object that is a long horizontal rod, but I have low confidence in my description.`"

This response illustrates that object O_B does not match the object structure S_A that the system was trained on. The system attempts to capture the presented object using its single-cylinder representation from E_{S_A}. However, the resulting reconstruction fails to match the observed object closely enough and the system indicates this misalignment with low confidence. Since it has only encountered single-cylinder objects during training, it cannot produce an accurate enough description of the two-part object with the current representation.

In a second interaction, the user defines an object structure S_B composed of two cylindrical components. Based on this specification, the system generates multiple prototype objects consisting of two cylinders with randomly sampled spatial arrangements, orientations, and sizes. After generating these prototypes and training the corresponding encoder model E_{S_B}, the user places a new object O_B in front of the sensor and again asks, "`What do you see?`" This time, the system utilizes the newly trained parametric shape encoder E_{S_B}, enabling it to produce an accurate description of the observed object:

> "`I see an unknown object, which consists of a long horizontal rod and a long vertical rod to the left.`"

This illustrates the system's ability to adapt its perceptual processing based on user input and to explicitly communicate uncertainty when its generated object representation, and consequently its description, does not adequately capture the observed object.

Note that the full data generation and encoder training runs entirely on the laptop's 12th Generation Intel Core i7-12800H central processing unit (CPU) in under 20 minutes.

A video demonstration of the experiments can be accessed at `https://tufts.box.com/v/ACSDemoVideo`.

6. Qualitative Results

We present additional qualitative results on recorded point clouds of two objects from the Yale-CMU-Berkeley (YCB) dataset (Calli et al., 2017), namely a skillet and a toy power drill, and a 3D replica of a screw object from the FetchIt! Challenge (Han et al., 2020). For the toy power drill, the system was trained on object structure S_{III}, consisting of two rod components with unconstrained positions and orientations. For the skillet and screw objects, the system was trained on object structure S_{V}, comprising one rod and one disk with unconstrained positions and orientations. Table 3 shows RGB images of the objects, visual descriptions inferred from the superellipsoid parameters, Chamfer distances between reconstructed and ground-truth point clouds, and generated human-language descriptions.

The Chamfer distances are computed within the reference cube, defined as the 3D region $[-1, 1]^3$, as described in Sec. 4.3. This ensures a consistent scale for computing Chamfer distances. Overall, the system captures the primary geometric features of all objects. Finer details, such as the drill's screw attachment and push button or the skillet lid's handle, are not explicitly captured by the superellipsoid-based model. This is expected, as the training prototypes do not include such fine-grained elements. For more complex objects, the system produces simplified approximations, effectively generating a coarse-grained representation. For instance, the skillet lid handle and the drill's button and screw attachment are represented as part of an "enveloping rod component."

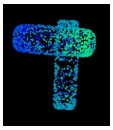
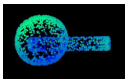
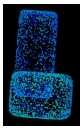
The Chamfer distances range from 0.029 to 0.076. These small values indicate that the system accurately captures the fundamental geometry of the objects. Slightly higher distances occur in regions corresponding to fine-grained features, such as the drill's screw attachment or push button, which are simplified in the superellipsoid reconstruction.

7. Conclusion

In this work, we address the challenge of enabling cognitive architectures to adapt their visual processing in dynamic environments. We extend the DIARC cognitive robotic architecture with a mechanism that dynamically adapts its perceptual vocabulary based on task-specific user instructions. By leveraging natural-language descriptions of object characteristics, the system synthesizes object prototypes that capture the intended structure.

These prototypes enable DIARC to autonomously train a parametric shape encoder, which infers geometric representations from depth observations of objects. The system establishes a bidirectional mapping between superellipsoid-based object representations and human language. User instructions are translated into mereological representations, while the inferred representations can be converted back into human-readable descriptions or schematic visualizations. This bidirectional mapping supports transparent and adaptive interaction with human users.

Table 3. Visual and human-language description for four different objects.

Object	**Toy Power Drill**	**Skillet**	**Screw 1**	**Screw 2**
Object (RGB)				
Object Structure Instructions	$HasComp(O_{\mathrm{III}}, C_{1,\mathrm{III}})$ $HasComp(O_{\mathrm{III}}, C_{2,\mathrm{III}})$ $Cylinder(C_{1,\mathrm{III}})$ $Cylinder(C_{2,\mathrm{III}})$	$HasComp(O_{\mathrm{V}}, C_{1,\mathrm{V}})$ $HasComp(O_{\mathrm{V}}, C_{2,\mathrm{V}})$ $Cylinder(C_{1,\mathrm{V}})$ $Disk(C_{2,\mathrm{V}})$	$HasComp(O_{\mathrm{V}}, C_{1,\mathrm{V}})$ $HasComp(O_{\mathrm{V}}, C_{2,\mathrm{V}})$ $Cylinder(C_{1,\mathrm{V}})$ $Disk(C_{2,\mathrm{V}})$	$HasComp(O_{\mathrm{V}}, C_{1,\mathrm{V}})$ $HasComp(O_{\mathrm{V}}, C_{2,\mathrm{V}})$ $Cylinder(C_{1,\mathrm{V}})$ $Disk(C_{2,\mathrm{V}})$
Visual Description				
Chamfer Distance	0.076	0.041	0.042	0.029
Human-language Description	I see an unknown object that consists of a long vertical rod and a long horizontal rod to the right on top.	I see an unknown object that consists of a long horizontal rod and a disk to the left.	I see an unknown object that consists of a long vertical rod and a disk at the bottom.	I see an unknown object that consists of a disk with a long vertical rod at the bottom.

We demonstrate the feasibility of our approach in an experimental scenario, illustrating how the system leverages user instructions to expand its object description capabilities. Additionally, we show how the system communicates uncertainty when its internal object representation does not adequately match an observed object, indicated by a low-confidence statement.

8. Limitations and Future Work

Our mapping from continuous superellipsoids to discrete semantic predicate labels is inherently affected by the vagueness and context-dependence of human language (Lim & Wu, 2023). For instance, different users may interpret the same shape labels differently, some shapes may be appropriate in certain contexts but not others, and some shapes may fall between categories. A promising direction for future work is to replace discrete predicate assignments with a continuous assessment of predicate applicability, enabling the system to generate more nuanced descriptions.

Objects often contain details at multiple scales. For example, the screw attachment of a toy power drill is currently represented simply as part of the overall rod structure. Our current object prototypes are relatively simplified, capturing an “enveloping representation" of the object. While this approach works well for a coarse-grained descriptions, it can over-simplify certain objects by omitting distinctive features. Nevertheless, the modular design of the proposed system is highly extensible and could be adapted to process components at different scales recursively, enabling more detailed representations in future work.

Currently, the system relies on meaningful user input to define the structure of a new object. A more user-friendly future direction would be to develop a workflow in which the system leverages its existing vision processing models to analyze the point cloud of a novel object and propose candidate superellipsoidal representations. The user could then review and validate these suggestions, selecting the most appropriate representation. Furthermore, the reconstruction error between the original point cloud and the reconstructed superellipsoid-based point cloud could be exploited to suggest new object structures or to guide the automated generation of prototype objects.

References

Alaniz, S., Mancini, M., & Akata, Z. (2023). Iterative superquadric recomposition of 3d objects from multiple views. *IEEE/CVF International Conference on Computer Vision*.

Barr, A. H. (1981). Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications*, *1*, 11–23.

Blender Online Community (2018). *Blender - a 3d modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam. From `http://www.blender.org`.

Boggs, J. (2025). Towards visual-symbolic integration in the soar cognitive architecture. *Cognitive Systems Research*, *91*, 101353.

Boult, T., et al. (2021). Towards a unifying framework for formal theories of novelty. *AAAI Conference on Artificial Intelligence*, *35*, 15047–15052. From `https://ojs.aaai.org/index.php/AAAI/article/view/17766`.

Boult, T. E., Cruz, S., Dhamija, A., Gunther, M., Henrydoss, J., & Scheirer, W. (2019). Learning and the unknown: surveying steps toward open world recognition. *Thirty-Third AAAI Conference on Artificial Intelligence*. AAAI Press.

Bulzan, A.-S., & Cernazanu-Glavan, C. (2025). Towards open world detection: A survey. From `https://arxiv.org/abs/2508.16527`.

Calli, B., Singh, A., Bruce, J., Walsman, A., Konolige, K., Srinivasa, S., Abbeel, P., & Dollar, A. M. (2017). Yale-cmu-berkeley dataset for robotic manipulation research. *The International Journal of Robotics Research*, *36*, 261–268.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*, 273–297. From `https://doi.org/10.1023/A:1022627411411`.

Cruz, S., Doctor, K., Funk, C., & Scheirer, W. (2025). Open issues in open world learning. *AAAI AI Magazine*, *46*.

Dubuisson, M.-P., & Jain, A. (1994). A modified hausdorff distance for object matching. *12th International Conference on Pattern Recognition* (pp. 566–568 vol.1).

Elmoznino, E., Jiralerspong, T., Bengio, Y., & Lajoie, G. (2025). Towards a formal theory of representational compositionality. *Forty-second International Conference on Machine Learning*. From `https://openreview.net/forum?id=fXCfT7ErvL`.

Fedele, E., Sun, B., Guibas, L., Pollefeys, M., & Engelmann, F. (2025). SuperDec: 3D Scene Decomposition with Superquadric Primitives. *IEEE/CVF International Conference on Computer Vision*.

Gao, Z., Yi, R., Huang, Y., Chen, W., Zhu, C., & Xu, K. (2024). Learning part-aware 3d representations by fusing 2d gaussians and superquadrics. *CoRR*, *abs/2408.10789*. From `https://doi.org/10.48550/arXiv.2408.10789`.

Goel, S., et al. (2024). A neurosymbolic cognitive architecture framework for handling novelties in open worlds. *Artificial Intelligence*, *331*, 104111. From `https://www.sciencedirect.com/science/article/pii/S000437022400047X`.

Han, Z., Allspaw, J., LeMasurier, G., Parrillo, J., Giger, D., Ahmadzadeh, S. R., & Yanco, H. A. (2020). Towards mobile multi-task manipulation in a confined and integrated environment with irregular objects. *2020 IEEE International Conference on Robotics and Automation (ICRA)* (p. 11025–11031). IEEE. From http://dx.doi.org/10.1109/ICRA40945.2020.9197395.

Holder, L., Langley, P., Loyall, B., & Senator, T. (2025). Introduction to open-world ai. *Artificial Intelligence*, (p. 104393). From https://www.sciencedirect.com/science/article/pii/S0004370225001122.

Jiang, S., Zhao, Q., Rahmani, H., Soh, D. W., Liu, J., & Zhao, N. (2024). Gaussianblock: Building part-aware compositional and editable 3d scene by primitives and gaussians. *CoRR*, *abs/2410.01535*. From https://doi.org/10.48550/arXiv.2410.01535.

Joseph, K. J., Khan, S., Khan, F. S., & Balasubramanian, V. N. (2021). Towards open world object detection. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 5830–5840).

Kara, S., Ammar, H., Denize, J., Chabot, F., & Pham, Q.-C. (2024). Diod: Self-distillation meets object discovery. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 3975–3985).

Kingma, D., & Ba, J. (2014). Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Kirk, J. R., & Laird, J. (2019). Learning hierarchical symbolic representations to support interactive task learning and knowledge transfer. *International Joint Conference on Artificial Intelligence*.

Kotseruba, I., & Tsotsos, J. K. (2020). 40 years of cognitive architectures: core cognitive abilities and practical applications. *Artificial Intelligence Review*, *53*, 17–94. From https://doi.org/10.1007/s10462-018-9646-y.

Lim, W., & Wu, Q. (2023). Vague language and context dependence. *Frontiers in Behavioral Economics*. From https://www.frontiersin.org/articles/10.3389/frbhe.2023.1014233/full.

Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*, 91–110.

Ma, T., Wang, Z., Zhou, J., Wang, M., & Liang, J. (2024). Glover: Generalizable open-vocabulary affordance reasoning for task-oriented grasping.

Madl, T., Franklin, S., Chen, K., Montaldi, D., & Trappl, R. (2016). Towards real-world capable spatial memory in the lida cognitive architecture. *Biologically Inspired Cognitive Architectures*, *16*, 87–104. From https://www.sciencedirect.com/science/article/pii/S2212683X16300135.

Mullappilly, S. S., Gehlot, A. S., Anwer, R. M., Khan, F. S., & Cholakkal, H. (2024). Semi-supervised open-world object detection. *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*. AAAI

Press. From https://doi.org/10.1609/aaai.v38i5.28227.

Neuwirth-Trapp, M., Bieshaar, M., Paudel, D. P., & Gool, L. V. (2025). Rico: Two realistic benchmarks and an in-depth analysis for incremental learning in object detection. *arXiv*.

Pietikäinen, M. (2005). Image analysis with local binary patterns. *Image Analysis* (pp. 115–118). Berlin, Heidelberg: Springer Berlin Heidelberg.

Potapov, A., Zhdanov, I., Scherbakov, O., Skorobogatko, N., Latapie, H., & Fenoglio, E. (2018). Semantic image retrieval by uniting deep neural networks and cognitive architectures. *Artificial General Intelligence* (pp. 196–206). Cham: Springer International Publishing.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 779–788).

Reynolds, D. A. (2018). Gaussian mixture models. *Encyclopedia of Biometrics*. From https://api.semanticscholar.org/CorpusID:1063711.

Ruan, X., & Tang, W. (2024). Fully test-time adaptation for object detection. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (pp. 1038–1047).

Scheutz, M., Krause, E., Oosterveld, B., Frasca, T., & Platt, R. (2017). Spoken instruction-based one-shot object and action learning in a cognitive robotic architecture. *16th International Conference on Autoomous Agents and Multiagent Systems*.

Scheutz, M., Williams, T., Krause, E., Oosterveld, B., Sarathy, V., & Frasca, T. (2019). *An overview of the distributed integrated cognition affect and reflection diarc architecture*, (pp. 165–193).

Sinha, S., Premsri, T., & Kordjamshidi, P. (2024). A survey on compositional learning of ai models: Theoretical and experimental practices. From https://arxiv.org/abs/2406.08787.

Trafton, J. G., Hiatt, L. M., Harrison, A. M., Tamborello, F. P., Khemlani, S. S., & Schultz, A. C. (2013). Act-r/e: an embodied cognitive architecture for human-robot interaction. *Journal of Human-Robot Interaction*, *2*, 30–55.

Wang, Y., Chen, L., Zhao, T., Zhang, T., Wang, G., Yan, L., Zhong, S., Zhou, J., & Zou, X. (2025). High-dimension prototype is a better incremental object detection learner. *The Thirteenth International Conference on Learning Representations*. From https://openreview.net/forum?id=6T8czSBWce.

Yoo, J., Lee, D., Chung, I., Kim, D., & Kwak, N. (2024). What, how, and when should object detectors update in continually changing test domains?

Zhang, W., Goodwill, J., Chase, T., & Marshall, J. (2025). Evaluation and integration of yolo models for autonomous crater detection.

Zhao, X., Ding, W., An, Y., Du, Y., Yu, T., Li, M., Tang, M., & Wang, J. (2023). Fast segment anything.