Neuro-Symbolic Robotics: Capabilities, Limitations, and Challenges

Emre Ugur[1,2], Alper Ahmetoglu[3], Yukie Nagai[4], Tadahiro Taniguchi[5], Matteo Saveriano[6], and Erhan Oztop[7,8]

[1]Middle East Technical University [2]Bogazici University [3]Brown University [4]The University of Tokyo [5]Kyoto University [6]University of Trento [7]The University of Osaka [8]Ozyegin University

***Abstract*—This paper examines the emerging field of Neuro-Symbolic Robotics by systematically mapping the strengths and limitations of existing neuro-symbolic robotic systems onto a novel taxonomy. Our taxonomy is motivated by a core question in cognitive and developmental robotics: how should neural and symbolic processes be organized to produce robots that can learn, develop, and reason as intelligent agents do, given that symbols are not passively received but actively constructed through embodied sensorimotor experience. With this we aim to identify open challenges and promising research directions toward robust, fully autonomous, and reasoning-capable robots. The advancements in computational power, robust neural structures, and extensive data have positioned Neural Networks as the preferred solution for robotic challenges involving emergent behavior, learning, adaptation, and, more recently, reasoning and communication. Despite these strengths, the deployment of robots in real-world settings requires properties such as verifiability, explainability, and interpretability, which Neural Networks lack. Furthermore, neural network-based models often struggle with generalization and extrapolation, thus restricting their use. Historically, symbolic systems have been integral to intelligent robotics due to their verifiability, explainability, and scalability; however, their manually programmed frameworks often fail to effectively manage the complexity and diversity of the robot's continuous and high-dimensional environments. In this paper, by examining the robotic works that combine, in a variety of patterns, neural networks with symbolic systems, we classify them into four main categories of intertwined (A), coupled (B), neuro-symbolic translation (C), and non-uniform neuro-robotic systems (D), and provide an in-depth analysis to elicit their potential for the future of robotics. Based on this analysis, we conclude that coupled neuro-symbolic robotic approaches offer the strongest capabilities for robot control and motion planning, and hence represent the most capable paradigm for addressing the breadth of real-world robotic challenges. Neuro-symbolic translation-based robotic studies, by naturally integrating safety into learning, provide formal verification and correctness guarantees, albeit currently having limited practical applicability. Finally, intertwined neuro-symbolic robotics adopts a cognitively inspired, developmental perspective to bridge neural and symbolic paradigms; yet, despite its transformative potential, its capacity to trigger an architectural paradigm shift has yet to be demonstrated.**

***Index Terms*—Neuro-Symbolic AI, Neuro-Symbolic Robotics, Cognitive Robotics, Symbol Grounding, Symbol Emergence, Explainable AI, Explainable Robotics, Neural-Symbolic Integration**

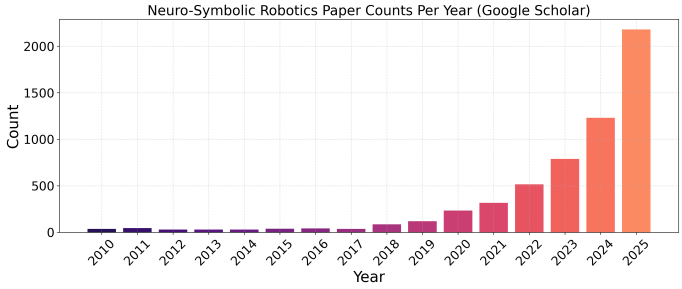


Fig. 1. Neuro-Symbolic Robotics related publication histogram, obtained via Semantic Scholar scraping. The data shows significant growth from 2021.

I. Introduction

Intelligent Robotics is characterized by the convergence of artificial intelligence and robotics, focusing on the creation of machines that can independently learn, reason, and execute complex tasks in ever-changing environments. The primary challenges in this field involve developing representations that are robust yet adaptable, thereby enabling high performance in complex tasks while also allowing for reasoning that ensures safety and explainability.

In contrast to domain-specific machine learning problems, such as classification and regression, a robot must process a continuous stream of sensorimotor data while acting in a world that is structured as a network of discrete entities with a range of relations among them. Thus, intelligent robots require cognitive mechanisms to process continuous sensory input and abstract symbolic structures [1]. In the following, we first address them separately under *Learning Systems*, and *Symbolic Systems* and later discuss how the two worlds can be intertwined together[1].

Learning Systems. The groundbreaking advances in deep neural networks and their effective application in artificial intelligence raise the question of whether deep learning with

Manuscript received December 19, 2025; revised May 20, 2026. This work was supported by the INVERSE project (101136067) funded by the European Union; by the World Premier International Research Center Initiative (WPI), MEXT, Japan; by the Scientific and Technological Research Council of Turkey (TUBITAK) ARDEB 1001 Program (124E227); and by the JSPS KAKENHI Grants 26K02973 JP25H01236. The authors thank Tuba Girgin for the illustrations.

[1]The term *symbol* is commonly used in the fields of AI and robotics within the context of symbolic AI, which is influenced by the physical symbol system hypothesis [2]. Here, we primarily use the term *symbol* in this sense. As Steels distinguished between m-symbols and c-symbols, the term *symbol* is often used with two fundamentally different meanings [3]. While this distinction is crucial when addressing symbols and meaning-making in cognitive science and semiotics, in this paper, we primarily focus on engineering problems in the field of robotics. For a more detailed discussion, please refer to [4].

big data is the solution that Intelligent Robotics has been seeking. The recent state-of-the-art robotic learning studies employ some form of (deep) neural network architecture, delivering superior performance compared to earlier traditional machine learning methods. Yet, the lack of transparency in neural networks poses serious concerns about their reliability, robustness, and safety [1].

Another significant criticism pertains to the brittleness (being susceptible to adversarial attacks) of the models [5], as well as the data efficiency and computational expense associated with robotic implementations [6]. While the computing cost can potentially be mitigated by suggesting a central pre-training that is performed once and subsequently deployed across various locations with minimal or no further training, addressing the black-box issues remains challenging because post-hoc explanations of neural network outputs may lack the reliability needed to persuade end-users to integrate these technologies into actual robots. For example, although recent Large Language Model (LLM) based systems enable pseudo-reasoning, their reasoning capabilities are not verifiable or reliable [7], even though they may be optimized through data fine-tuning and/or reinforcement learning with human-annotated reward labels for valid chain-of-thought generation. Thus, in intelligent robotics, the reasoning capabilities of these systems are limited to constrained laboratory settings.

Computationally, neural networks can represent propositional logic and a restricted subset of first-order logic, but cannot represent the full functionality and representational power of first-order logic, according to Marcus [6]. Thus, although LLMs possess impressive reasoning-like abilities, it is unclear how formally defined computational semantics can be neurally embedded in the operation of LLMs to address questions of reliability, trustworthiness, and safety in robotics.

Symbolic Systems. Symbolic systems are reliable in terms of planning and reasoning abilities, as computation steps can be explained and proved for correctness [8], [9]. However, they lack flexibility [10]. For example, an execution plan can be correct but still may fail in a given environment setting if the symbols used to capture the current setting lack the required resolution or sensitivity. This is a classic example of the problem of pre-defining a set of symbols and rules to represent the sensorimotor experience of a robot, often referred to as the *symbol grounding problem* [11]. Regardless of how well a symbolic system may be designed, it inevitably becomes fragile when confronted with minor alterations in the embodiment, environment, or task that were not anticipated during the design phase. Another issue with symbolic systems is that they restrict reasoning to symbolic space. However, the robotics tasks of planning, monitoring, and validation may require representations at multiple levels of abstraction beyond a single symbolic level. Although the levels of abstraction used in robotics literature are delineated within the context of natural language representation in [10], a theory of multi-level symbolic manipulation that bridges low-level sensory input with higher-level representations is lacking [5].

Neuro-Symbolic Systems. With the recent advances in deep neural networks, it has become more possible to address the symbol grounding problem by letting advanced neural network architectures learn symbolic representations. These representations can be used not only in symbolic manipulation systems for reasoning and planning, but also to equip the robot with various capabilities or enhance existing ones. With these advances, the number of neuro-symbolic studies has significantly increased in recent years, as shown in Fig.1: The number of neuro-symbolic robotics publications grew from approximately 200 per year in 2019 to over 2,000 per year in 2024, representing a 10-fold increase over five years, with the steepest growth occurring after 2021.

Although there are very preliminary efforts for proposing general paths for adapting neuro-symbolic approaches into, for example, industrial [12], surgical [13], or assistive [14] robotics, a general overview of the current studies and a possible generic architecture that utilizes the full power of both neural and symbolic systems is still missing. In the remainder of the paper, we present an analysis of key robotic works that utilize symbolic and learning systems in varying combinations.

The remainder of this paper is organized as follows. Section II describes the methodology and search strategy used to identify the reviewed studies. Section III presents the proposed taxonomy. Sections IV through VII analyze each of the four main categories in detail. Section VIII provides a general discussion, and Section IX concludes the paper.

II. Methodology

Search Strategy: Expert-driven selection and Google Scholar is used as the database to identify the papers in this manuscript. Our search included the keywords “neural networks”, “neural”, “learning”, “symbols”, “symbolic”, “rule”, “robots”, and “robotics”. We selected papers, including those from simulated or real robot experiments, that appear in known venues. Additionally, to ensure comprehensiveness, we reviewed surveys on symbol emergence and grounding, as well as neuro-symbolic AI, even though they may not directly address robotics. Finally, we also included some non-robotic neuro-symbolic studies and non-neural robotic studies that serve as a conceptual background for the studies reviewed.

It should be noted that our search strategy follows an expert-driven selection process rather than an exhaustive retrieval protocol. The goal was to identify representative and influential works that illuminate the key architectural patterns in the field consistent with the narrative review methodology[2], where the quality and coherence of the conceptual synthesis, rather than the exhaustiveness of retrieval, is the primary measure of rigor. For the same reason, this survey adopts a qualitative synthesis

[2]This work does not adopt a strict systematic review methodology [15], as its primary goal is not to exhaustively retrieve and evaluate all relevant studies under a predefined protocol, but rather to provide a structured and conceptual synthesis of the field. In particular, the rapidly evolving and interdisciplinary nature of neuro-symbolic robotics, spanning robotics, machine learning, and cognitive science, makes it difficult to define a closed set of search queries and inclusion criteria that would comprehensively capture the literature. Moreover, the lack of standardized benchmarks, consistent comparison metrics, and uniformly reported evaluation protocols across studies further limits the feasibility of systematic aggregation and objective comparison. Instead, this survey follows an expert-driven selection process, focusing on representative and influential works to develop a coherent taxonomy, highlight key challenges, and identify emerging research directions.

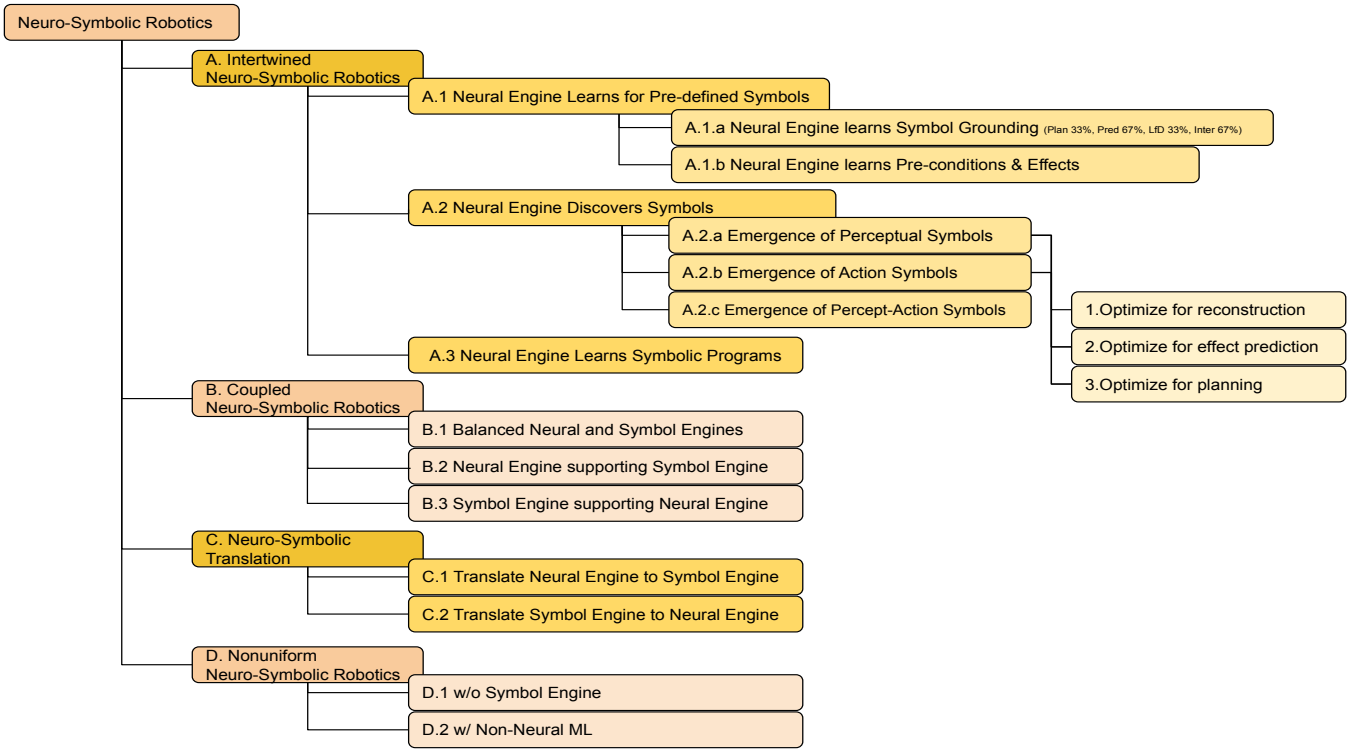


Fig. 2. A taxonomy of Neuro-Symbolic Robotics

TABLE I
DEFINITIONS

Symbolic Representation: Discrete encoding of the perception or action of the robot or the state of the environment.
Continuous Representation: Continuous encoding of the perception or action of the robot or the state of the environment.
Symbol Engine: The methods and algorithms used for manipulating symbols.
Neural Engine: Neural Network model used in discriminative or generative tasks.

rather than a quantitative meta-analysis: given the field's early-stage development (Fig. 1), the reviewed studies span highly heterogeneous setups, platforms, and evaluation protocols, and standardized performance metrics such as task success rates, mean-time-to-failure, or computational overhead are not consistently reported across studies, making aggregation across incompatible measures potentially misleading. Of the reviewed studies, 87% appear in peer-reviewed venues, of which 73% are conference papers and 27% are journal articles. Still, we note that the reviewed studies span a 25-year period from 2000 onwards, though 95% of them are concentrated in the more recent [2011 -] window, reflecting the field's rapid growth following the deep learning revolution.

Definitions: The analysis presented in this article is guided by our proposed taxonomy, as shown in Fig. 2, which categorizes robotic studies into four main groups based on the interplay between the neural and symbolic engines used. The phrases of *symbolic* and *continuous representations*, as well as the terms *symbol* and *neural engines*, are defined in Table I. Note that the discrete or continuous nature of an input or output does not, by itself, determine whether a computational pipeline should be considered symbolic or neural. As such, both discrete and continuous representations can serve as inputs and/or outputs of either a symbolic engine or a neural engine.A Symbol Engine might correspond to a classifier, such as a decision tree, a Monte Carlo search tree used for multi-step prediction, a set of operations over Domain-Specific Languages (DSLs), or a full-fledged off-the-shelf AI planning system in standard symbolic languages, such as the Planning Domain Description Language (PDDL). Likewise, the inputs and outputs of the Neural Engine, as well as the intermediate representations, may be discrete or continuous-valued, depending on the task.

III. A Taxonomy of Neuro-Symbolic Robotics

The proposed taxonomy is motivated by a core question in cognitive and developmental robotics: *how should neural and symbolic processes be organized to produce robots that can learn, develop, and reason in the way intelligent agents do?* In developmental cognitive systems, a central insight is that symbols are not passively received but actively constructed through embodied sensorimotor experience a perspective that sits at the heart of this taxonomy. Unlike prior Neuro-Symbolic AI taxonomies that abstract away from embodiment, robotics demands that symbolic representations be grounded in continuous sensorimotor experience, which fundamentally shapes how Neural and Symbol Engines must interact. The taxonomy is therefore structured along three developmentally and cognitively meaningful axes: (1) the degree of architectural coupling between engines whether they are inseparable, mutually supportive, or mutually transformable; (2) the direction of information flow between neural and symbolic processes; and

(3) the degree of autonomy in symbol acquisition ranging from systems that rely on manually predefined symbols to those that discover symbols autonomously through unsupervised robot interaction, analogous to the progressive symbol formation observed in developing agents. Together, these axes provide a principled, embodiment-aware framework for evaluating each category across the manifolds of control, planning, safety, perception, and real-world applicability. The taxonomy was developed through an iterative, expert-driven process, where categories emerged from repeated passes through the literature and were refined until until a primary category could be assigned to each reviewed study; we acknowledge that future work could further validate this categorization through formal inter-rater reliability assessment [16].

It should be noted that the proposed taxonomy represents regions on a spectrum rather than rigid, mutually exclusive partitions. In practice, some systems exhibit characteristics of more than one category, and their classification may depend on implementation details. For instance, a system in which a neural perception module produces symbolic predicates consumed by a planner could be interpreted as either intertwined (A) or coupled (B), depending on whether the symbolic planner retains independent operational capability. Similarly, translation-based methods (C) that refine symbolic policies through neural learning share structural features with Symbol Engine supporting Neural Engine approaches (B.3). In such borderline cases, we classify systems according to the *primary architectural intent*: whether the system is designed around the dependency between engines (A), their modularity (B), or the transformation from one representational regime to another (C). We acknowledge that such boundaries are inherently gradual, and encourage future work to revisit and refine these distinctions as the field matures.

In our taxonomy (Fig. 2), the first category involves neural and symbolic engines that are intertwined in such a way that the system can only operate when both engines are running. Most methods in this category utilize the neural engine to generate representations that are used by the symbolic engine. Next are the coupled neuro-symbolic methods in which the main engine, either neural or symbolic, is supported by the other to arrive at a decision. The core characteristic of this group is that the main engine can operate independently; running both engines typically enhances overall performance. Methods in the third category are translation-based systems that either translate symbolic rules into neural-processable data or distill information in a neural engine into symbolic knowledge, allowing for verifiability and interpretability. The methods falling into the fourth category do not possess a neural or symbolic framework that would qualify them as genuine neuro-symbolic methods. Still, they are included in this review because they are precursors to their neural counterparts and due to their potential for adaptation into such methods.

IV. A. Intertwined Neuro-Symbolic Robotics

In this category (Fig. 3), the representations, rules, or programs used by the Symbolic Engine are generated by the Neural Engine. A key distinction among these approaches is whether symbol or program learning is at the core and whether

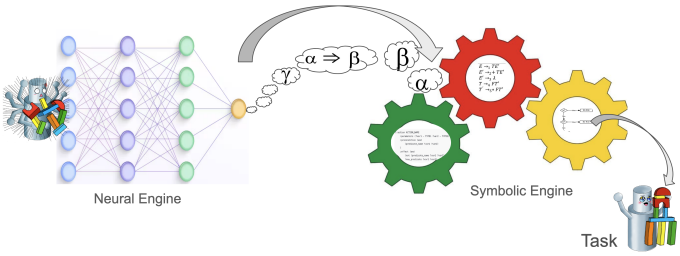


Fig. 3. The relation between Neural and Symbolic Engines in Intertwined Neuro-Symbolic Robotics (A). The robot's continuous interaction experience is used by the Neural Engine to learn discrete symbols and/or rules and propagated to the Symbolic Engine for high-level reasoning.

it is undertaken by the neural system. Accordingly, we have three main subcategories, which are detailed next.

A.1 Neural Engine Learns for Pre-defined Symbols

In this sub-category, the symbols and the operators used by the Symbol Engine are predefined based on the task and domain requirements. The Neural Engine either learns the mapping between these discrete symbols and the robot's continuous sensorimotor experience or the set of symbolic preconditions and effects of the operators derived from the robot's experience.

A.1.a Symbol grounding: For constrained domains and tasks, planning is possible with pre-defined sets of predicates, operators, and pre-conditions and effects of these operators. In these situations, the representational gap between the symbols and the continuous representation the robot faces should be addressed. For this, in one study, given pre-defined predicates in pre- and post-conditions of manually designed transition rules, the robot learned the mapping from its own percepts to the corresponding predicates' post-conditions using kernel perceptrons [17]. More recently, Yuan et al. learned to process RGB image patches conditioned on canonical object views into embeddings that can be classified into single and relational object-object logical predicates encoded in action preconditions from demonstrations, to be used in planning [18]. Given a robot interaction video dataset with annotated actions and manually implemented preconditions and effects, trained classifiers were used to map the bounding boxes of objects to the corresponding symbols [19]. When applicable, the approach taken in this category is effective; however, the fixed set of predicates and predefined rigid rules and operators limits the practical deployment of the method to well-structured environments and tasks. In the next section, we provide an overview of the methods that relax this restriction by learning a set of predicates for planning operators.

A.1.b Pre-condition and effect learning: The rigidity in the predefined transition rules can be relaxed by learning the set of predicates for the preconditions and postconditions of the operators used by the Symbol Engine. In this category, the mapping between the sensorimotor space of the robot and the symbols has been established previously, and the robot learns the set of predicates for the preconditions and effects of actions. Diehl et al. learned a list of preconditions and effect predicates, first segmenting the human demonstrations into actions, then extracting the relevant preconditions and

Dimensions: How the Sampled Studies Contribute
Control: Not the primary focus. Predictive policies were learned from observations [18], [23]; DMPs were learned via imitation learning [22].
Motion Planning: Not explicitly addressed. DMP outputs generated motion trajectories [27]; an optimization procedure formed collision-free expert trajectories [23].
Safety: None of the studies explicitly addressed safety (safety guarantees, safe exploration, or formal verification) as a primary design concern.
Architectures: Pipelines integrating grounding, demonstration parsing, and operator extraction [20], [24].
Perception: Strong perceptual processing. Grounding methods [17], [18] transformed high-dimensional visual observations into symbolic representations, mapping perceptual features to symbolic preconditions/effects [19], making perception directly usable by planners.
Applications: Tabletop manipulation [22], action sequencing [20], and planning from annotated videos [19].

TABLE II
CONNECTION BETWEEN ROBOTICS AND THE STUDIES CATEGORIZED AS INTERTWINED WHERE PRE-DEFINED SYMBOLS ARE LEARNED (A.1).

postconditions based on a counting heuristic, and finally generating the related planning operators [20]. Ultimately, an externally defined goal can be achieved through a sequence of actions using the Fast Downward PDDL planner [21]. Ahmadzadeh et al. learned action pre-conditions and effects in the form of lists of symbolic predicates from provided human demonstrations in the manipulation domain and verified through the PDDL planner [22]. Given predefined symbolic predicates, Mukherjee et al. learned a set of parameterized actions, along with their corresponding precondition and effect predicates, in a manipulation domain ([23]). Gugliermo et al. proposed learning planning domains from observed traces using Behavior Trees as intermediate, human-readable structures [24]. Given a symbolic goal, the symbolic operators were learned to synthesize a plan and its low-level controller in an RL framework [25]. Lay et al. used preconditions and effect symbols to detect task-specific deficiencies and support humans in action feasibility, rather than plan generation [26]. These studies are constrained to a predefined set of symbols, whereas robots may need to invent new symbols in changing environmental conditions and new environments. We recap the contributions of the systems to robotics, where the Neural Engine learns for predefined symbols (A.1), and then cover the neuro-symbolic studies that address the unsupervised or self-supervised discovery of such symbols in the next section.

Contributions to Robotics: The contributions of the systems categorized under A.1 to various dimensions of robotics are summarized in Table II, together with representative studies. Rather than directly addressing control and motion planning, these studies focused on the perceptual grounding of predefined symbols in the continuous sensorimotor world of a robot; therefore, precise control or motion planning algorithms were not offered in these studies. Similarly, although capabilities suitable for safety-related use have been developed, no direct safety applications have been demonstrated. These methods offer robust and precise perceptual processing capabilities, enabling strong mappings between high-dimensional visual observations and their corresponding symbolic representations, as well as components of operator preconditions and effects that can be utilized for planning. These systems are particularly well-suited to table-top manipulation and structured tasks [19], [22], but their dependence on the predefined predicates limits their suitability for the variability and unpredictability of real-world environments.

A.2. Neural Engine Discovers Symbols

In this section, we cover the studies where Neural Engine is used to discover perceptual (e.g., [28]), action (e.g., [29], [30]), or sensorimotor (e.g., [31]) symbols that are directly used by the Symbol Engines for planning and for other purposes.

A.2.a Perceptual Symbol Emergence (PSE): Neural Engines, in this category, form or discover symbols [32] in the continuous sensory/perceptual space of the robot. These symbols are typically employed as predicates in planning preconditions and post-conditions of action operators. These studies optimize the process of organizing continuous perceptual space to identify discrete symbolic categories using various approaches and metrics, as follows.

a) A.2.a.1 Optimize PSE for reconstruction: From the perspective of perceptual symbol systems [33], one of the primary purposes of symbolic representation is to enable the reconstruction of sensorimotor observations by structuring information into compact and meaningful representations. In particular, in multi-modal learning, a category inferred from one modality (e.g., vision) can be used to reconstruct another modality (e.g., haptics) through cross-modal inference. To minimize reconstruction error, probabilistic generative models (PGMs), which incorporate discrete categories as latent variables to predict multimodal observations, have been actively pursued from the early to mid-2010s [34]. These studies, in general, clustered sensorimotor data, formed categories, and connected these categories to symbolic representations. As systems for symbol emergence in robotics have become increasingly complex, a general framework for distributed development and integration (Neuro-)SERKET has been proposed, which is aimed at facilitating modular and scalable implementation of these systems [35], [36]. This approach has led to the concept of a large-scale cognitive architecture, namely a whole-brain probabilistic generative model [37].

The studies above focused on probabilistic learning and inference; however, they did not fully exploit Neural Engines within their frameworks. On the other hand, Ozturkcu et al. studied whether a deep reinforcement learning system could entail the development of high-level neural encodings that might be viewed as antecedents of symbolic representations [38]. They demonstrated that even without explicit design or engineering, neural responses resembling abstract symbol-like representations can emerge in their system. Recently, Gopalan et al. proposed first discovering skill segments from demonstration trajectories and then applying unsupervised clustering and SVM classification to identify and learn the mapping of the potential termination states of each learned skill [39]. Eventually, these were used to learn the relation between natural language sentences and sequences of the

learned abstract symbols using a Seq2Seq recurrent neural network. While these methods can categorize the continuous perceptual space of the robot into discrete symbols, whether such categorization is effective for downstream tasks has not been addressed in these studies. In the following subsections, we outline the methods that support the categorization process by considering the effectiveness of symbols in relation to the task goals.

A.2.a.2 Optimize PSE for action effect prediction: The previous unsupervised clustering approach identifies discrete symbols without guaranteeing their utility for the Symbol Engine. To address this problem, several research groups investigated methods for discovering symbols that are guaranteed to be useful for one-step action effect prediction, the most fundamental step in symbolic planning. Ahmetoglu et al. proposed and realized a general neural framework, namely DeepSym, that translates the robot's raw sensorimotor experience into the symbolic domain [28]. With this architecture, given a continuous interaction experience, the robot can discover object and effect symbols that can be automatically translated to Probabilistic Problem Domain Definition Language (PPDDL) for generating high-level symbolic plans. To achieve this, a predictive deep encoded-decoder network with a binary bottleneck layer was trained with initial and outcome scene images to extract action and effect-grounded object and outcome categories, which in turn were used to make single-step action predictions. As an intermediate step, a decision tree was constructed based on the interaction of the robot, now represented with the discovered symbols corresponding to object and effect categories. Subsequently, each path in the decision tree was converted to an operator in PPDDL format, allowing the use of off-the-shelf planners as Symbol Engines. The system was verified in a tabletop environment, where symbols such as pushable, rollable, or insertable were discovered and utilized to create effective plans, for example, to build towers of varying heights by manipulating objects within the environment. A limitation of this work is that it can generate symbols for single or pairs of objects. As such, preconditions and effects are constrained to planning operators involving single or paired object predicates. Yet, many complex actions involve interactions with varying numbers of objects, or the effects of actions influence multiple objects in environments such as cluttered or articulated settings. To address this issue, DeepSym was extended by incorporating a transformer-based structure [40] that learns which object features to attend to through a set of attention weights [41], [42]. As in DeepSym [28], these weights were channeled through a discrete activation layer that generates relational symbols, capturing the interactions in the environment. The discovered relational symbols between objects are then combined with the discovered single-object symbols to predict the outcomes of robot actions for the objects available to the robot. In the follow-up work, off-the-shelf planners were enabled by converting the discovered multi-object symbolic interactions into PDDL operators [43]. This study demonstrated that complex plans involving multiple objects can be generated using symbols discovered by the system, thereby addressing object affordances [44] to some extent; however, the generalization based on the affordances of complex-shaped objects has not been addressed.

A.2.a.3 Optimize PSE for planning: The symbols that are effective in single-step action prediction do not necessarily guarantee optimal planning performance. Therefore, it is essential to investigate methods for maximizing the utility of discovered symbols in enhancing long-horizon planning performance. In this vein, Weerapaneni et al. proposed learning abstract relational symbolic object representations from visual observations in an unsupervised manner and using them to create multi-step plans [45]. The Neural Engine output was used as the parameters of a Hidden Markov Model-like probabilistic model, which in turn could be used as the Symbol Engine. The model was used for model-based RL in simulated tower-building tasks with simple blocks, given the images of goal and initial block configurations. While this system could effectively create symbolic plans using the discovered object symbols, the method was verified in a simulation environment with abstracted robot actions, such as pick-and-place locations. In other words, it discarded the complex dynamics between the robot's embodiment, its interactions, and its physical environment.

A.2.b Action Symbol Emergence (ASE): The studies introduced thus far have focused on perceptual symbol formation and assumed an existing finite action repertoire. In a more general setting, lifelong cognitive robotic systems need to extend their action repertoire [46] through interactive learning. In this section, we review studies in which action symbols were discovered and utilized in Symbol Engines.

A.2.b.1 Optimize ASE for action effect prediction: Silver et al. proposed formalizing the operator learning problem within the Task and Motion Planning (TAMP) framework [30]. The proposed system learns operators from previously acquired action symbols, which can be defined as a lossy abstraction of the domain's transition model. Following [30], Chitnis et al. proposed neuro-symbolic relational transition models (NSRTs) in which a task plan skeleton is generated using a symbolic engine that describes the high-level transitions, and then the neural engine searches for low-level operator parameters [47]. If the plan skeleton is not *downward refinable*, i.e., if there is no parameterization of the lower-level skill that makes the plan successful, a new plan skeleton is generated. Hence, *bilevel* planning in both discrete and continuous levels was established, allowing the robot to make detailed plans considering the geometric information. However, NSRTs were learned from a given set of parameterized skills, whereas in [48], these skills were also learned from low-level demonstrations, providing a complete bilevel operator learning stack.

In [49], a hierarchical RL framework is adopted, and a diverse set of actions is discovered while simultaneously learning symbolic forward models through intrinsic motivation signals given pre-defined state abstractions. As the Symbol Engine, the system employed the breadth-first search method, where each expansion corresponded to the learned symbolic forward model and was executed sequentially to reach the goal. Wang et al. proposed to learn constraints that address the effectiveness of actions using Gaussian Processes

Dimensions: How the Sampled Studies Contribute
Control: Not explicitly addressed. An exception: discovered discrete actions were parameterized via gradient-descent optimization in [60].
Motion Planning: Not explicitly addressed, but classical planning was. Abstract relational symbols [45] supported multi-step planning via HMM-like models. Transformer-based multi-object operators [43] enabled longer-horizon classical planning.
Safety: Not explicitly studied, though highly relevant. Predictive symbol formation [28], [41] may restrict transitions to those satisfying estimated effects, reducing unsafe interactions.
Architectures: Probabilistic generative frameworks (MLDA, SERKET) [34]–[37] provide scalable pipelines for multimodal symbol emergence.
Perception: Strong perceptual methods. Cross-modal symbol emergence [34] created mutually predictive sensory categories. Relational attention learning [41], [42] extracted key perceptual features for planning.
Applications: Tower building [28], [45], multi-object manipulation [43], and natural-language-to-symbol pipelines [39].

TABLE III
CONNECTION BETWEEN ROBOTICS AND THE STUDIES CATEGORIZED AS INTERTWINED WHERE SYMBOLS ARE DISCOVERED (A.2).

[50], [51]. They employed a sampling method to generate a comprehensive set of potential action parameters, along with the associated skills. Given a goal and learned parametric motion primitives, the planning system receives perceptual state estimates from the Neural Engine to generate a plan using their so-called PDDLStream framework [52], [53]. Last but not least, Aktas et al. discovered action symbols from human demonstrations and exploited visual language models (VLMs) not only to label those actions but also to generate plans through their scene interpretation and reasoning capabilities [54].

A.2.b.2 Optimize ASE for planning: While the studies in the previous sections either used unsupervised clustering or self-supervision based on single-step effect prediction to learn predicates, symbolic predicates were learned using a surrogate objective for multi-step planning [55]. In this work, instead of the robot's own self-exploration experience, interactions obtained from demonstrations were used. In a follow-up work, Kumar et al. reduced the complexity of the learned operators by focusing on a subset of abstract effects [56]. These studies not only learn action symbols but also identify the motion parameters that enable TAMP. Although symbolic predicates are also learned, they are defined over already available high-level predicates, which may not be realistic to assume in lifelong scenarios. Following [55], Li et al. proposed to learn predicates by actively collecting information by querying an expert [57]. Finally, Wong et al. used LLMs in an interactive planning loop to jointly learn a set of symbolic action abstractions and their low-level controllers [58]. Following the TAMP formulation, [59] demonstrated that abstract model learning enhances the sample efficiency of planning and learning. Given action trajectories, they learned abstract Markov decision processes (MDPs) that encode states and temporal dimensions in higher granularity levels, optimizing their planning performance.

Studies mentioned in this section either assume the existence of high-level predicates or a set of demonstrations from which a good set of state symbols can be learned. As such, the low-level policies of operators are learned with supervision, either in the form of state symbols or demonstrations that solve the task. It is also worth mentioning option discovery methods [61]–[65] that focus on learning these low-level policies directly from raw sensory space by exploration. While these methods do not directly utilize a symbol engine, they provide a finite number of low-level policies with their initiation and termination conditions overlapped in the state space, which lays the groundwork for action and state symbol learning.

A.2.c Percept-Action Symbol Emergence (PASE): While previous studies focused on discovering either perceptual or action symbols, some recent studies have investigated the possibility of creating both perceptual and action symbols from the sensorimotor experience of the robot [31]. Shah et al. used critical regions [66], [67] and high-density regions over the state space as state abstraction targets [68], [69]. After learning the state abstractions, action abstractions were learned on top of them. To obtain the density distribution of a state space, low-level demonstrations generated by motion planners and controllers were used. More recently, other researchers [70], [71] extended [55] by first learning high-level predicates directly from raw state representations using visual language models (VLMs) and then learning operators defined over these predicates. Last but not least, the perceptual and action symbols were simultaneously learned by a predictive neural network inside its discrete bottleneck layer and used to make plans using a symbolic tree search strategy in single- and double-object manipulation tasks [60]. Importantly, this study employed an entropy-based curiosity mechanism to guide exploration and proposed a gradient-descent-based optimization process to convert symbols into continuous action parameters. We close this section by summarizing the robotic utility that can be provided by systems in which the Neural Engine discovers symbols (A.2).

Contributions to Robotics: The contributions of the systems categorized under A.2 to various dimensions of robotics are presented in Table III, together with representative studies. As in A.1 approaches, these methods have not explicitly addressed control or motion planning; however, the learned representations can be used for both, as they enable controllers and motion planners to reliably predict outcomes before actuation and plan execution. By enabling autonomous PDDL operator generation [28], [43] and unsupervised relational abstractions for long-horizon reasoning [45], these studies show strong long-horizon planning capabilities, although planning quality remains sensitive to discretization errors in symbol discovery. These methods rarely encode explicit safety constraints, meaning that although discovered symbols may capture affordances [44] that may enable safety checks, they often lack symbolic safeguards against hazardous or irreversible actions. In fact, predictive symbols and rules can be used to reduce the risk of unsafe interactions, for example, with probabilistic representations [28]. These studies are, in general, perception-rich, as demonstrated by multimodal PGMs [34] and transformer-based relational discovery [41] that derive symbols directly from sensorimotor data. Current application domains of these approaches cover table-top manipulation and structural pattern formation. The complexity and unpre-

Dimensions: How the Sampled Studies Contribute
Control: Not explicitly addressed except in [75], where the NSDP integrated filtering acceleration and logical operations.
Motion Planning: Not explicitly addressed, but symbolic planning was. LLM-generated PDDL-like planners [73] and behavior-tree conversion pipelines [72] supported long-horizon planning; motion planning was delegated to external libraries.
Safety: Logic rules and safety metrics were incorporated to achieve stable and safer behaviors [75]; runtime monitoring via precondition verification was also employed [73].
Architectures: Hybrid stacks combining LLMs, Prolog, PDDL, DSLs, and planners [72], [73].
Perception: Not the primary focus; LLM symbolic operators reference entities parsed by external perception modules.
Applications: Robot planning from instructions [72], [73], interactive teaching of predicate definitions [74], and interpretable autonomous driving [75].

TABLE IV
CONNECTION BETWEEN ROBOTICS AND THE STUDIES CATEGORIZED AS INTERTWINED WHERE SYMBOLIC PROGRAMS ARE LEARNED (A.3).

dictability of real-world manipulation remain challenging, as these methods primarily focus on perceptual and reasoning aspects of manipulation, while delegating control and motion planning to classical implementations.

A.3 Neural Engine learns Symbolic Programs

Within this category, all symbolic programs are crafted by the Neural Engine, predominantly using LLMs, and subsequently processed by the Symbol Engine. In [72], a Neural Engine (LLM) was used to produce parallel plans, which were then converted into behavior trees and executed by the robot. The conversion here used Prolog as a bridge between symbols given in natural language and the behavior trees. Capitanelli et al. [73] trained LLMs to produce neuro-symbolic task planners, which are consistent with PDDL syntax. With this, they achieved better scalability as the domain complexity increased. This approach also enabled the production and execution of actions without waiting for the whole plan to be generated. In [74], a pre-trained LLM as a Neural Engine was used to learn symbolic predicates, in the form of Python program segments, from human language feedback during robot interactions. Subsequently, symbolic operators were learned through a clustering algorithm, enabling plan generation.

Last but not least, Sun et al. [75] provided interpretability in symbolic decision-making in autonomous driving by integrating neural and symbolic approaches while also achieving safe and consistent behavior. For this, a Neural Engine was trained to select operations from a set of symbolic pre-defined operations. This allowed the generation of a sequence of operations given goals in Domain Specific Language (DSL), which was used by the Symbol Engine for planning and control. We expect to see more studies in this category with the recent advances in LLMs, where inputs in multiple modalities and user-specified structured outputs facilitate the interface between natural language and DSLs [7]. We wrap up this section by highlighting the key utilities that can be provided for robotics.

Contributions to Robotics: The contributions of the systems categorized under A.3 to various dimensions of robotics

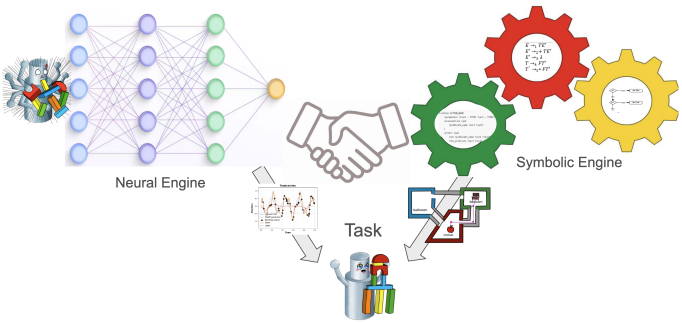


Fig. 4. The relation between Neural and Symbolic Engines in Coupled Neuro-Symbolic Robotics (B). In a nutshell, they can run independently and support each other for robot control.

are presented in Table IV, together with representative studies. As these studies focused on the high-level sequencing of logical actions or decision-making, they did not explicitly address continuous control or motion planning. To a large extent, the generation of motion trajectories and actions was delegated to other modules. An exception is Sun et al. [75], who explicitly integrated continuous parameters such as velocity, acceleration, and safety distance into the output of the decision module. Saccon et al. [72] performed temporal planning by first generating the plans using Prolog and converting them into partial-order plans using Simple Temporal Networks, whereas PDDLS are generated by LLMs in [73]. Safety was ensured when symbolic constraints restrict allowable behaviors (e.g., in driving scenarios [75]), yet the lack of formal guarantees keeps these models vulnerable to unsafe edge-case outputs. Perception grounding modules were typically provided externally. Overall, these systems show promise in robot planning-from-language tasks [72], [73] and safe-driving decision generation [75].

Discussion

The intertwined neuro-robotics studies (Fig. 2 (Category A) tightly integrate Neural and Symbol Engines, such that the Symbol Engines are designed to operate only with the representations or rules that are expected to be generated by the Neural Engines. In these studies, multi-step plans are generated using PDDL-like languages and executed in real or simulated robots, mostly in the manipulation domain. They mostly learn from interactions with the environment, whereas learning from demonstration or reinforcement learning approaches is used less frequently. The studies that learn predefined symbols or discover symbols do not generally extend beyond planning and are not used for natural language communication. The studies that learn symbolic programs, on the other hand, integrate recent LLM modules and, therefore, can address natural language. However, few of them discuss verifiability issues. Lastly, they primarily use a predefined set of symbols, which limits their broader applicability.

V. B. COUPLED NEURO-SYMBOLIC ROBOTICS

In this section, we provide an overview of the robotic systems where the Neural Engine and Symbol Engine modules interact with each other by combining their outputs or by

supporting one another (Fig. 4). However, they are not as tightly connected as in the intertwined approaches, i.e., they may work as separate modules for robot control.

B.1 Balanced Neural and Symbol Engines

Say-Can [76] has been one of the first studies that benefited from the reasoning capabilities of LLMs for robot control. Given a goal, Say-Can utilized a language model (PALM [77]), which provided high-level semantic knowledge about the task and generated a list of actions to achieve it. To ground the corresponding actions in the robot's actual world, an affordance-based value function module was implemented, which was used to weigh and filter the actions produced by the language model. This method can be considered the first LLM-based model capable of completing long-horizon natural language instructions by using a mobile manipulator. Neural and Symbol Engines were conceived as building blocks in [78], encompassing a modular approach where action primitives were defined to handle independent sub-tasks. The input query was processed by a language parser, transforming it into an executable program composed of these primitives. Note that some primitives were symbolic (e.g., counting), and others were implemented with neural networks (e.g., visual grounding). [79] used a Symbol Engine to select a list of actions, using First-Order Logic (FOL) rules that represent human background knowledge about the driving environment in each RL exploration step, and a Neural Engine, which approximates the Q value function, to select the action to execute following the learned policy. With this, they ensured safety and also enabled control in the continuous state and action space. In another study, [80] integrated logical rules, ontologies, and LLM-based planners, exploiting symbolic information to enhance the ability of LLMs to generate recovery plans. Given the instructions, their robot began executing the actions outlined in the plan generated by the LLM. The effect of each action was observed and provided as input to the sub-goal verifier, which used an ontology to decide whether the action was successful or not. In the event of failure, the ontology was used again to determine the recovery strategy, which was then provided to the LLM planner for replanning. In [81], human instructions were translated into executable robot plans by using LLMs to decompose the tasks into sub-goal descriptions that were executed by the planner sequentially. They used scene graphs as the intermediate representations. [82] combined symbolic and geometric scene graphs for vision-based long-horizon hierarchical planning. A symbolic scene graph was used to identify the next sub-goal from the goal description, and a geometric scene graph was utilized to predict the motion parameters based on the objects embedded within it.

B.2 Neural Engines supporting Symbol Engines

In this category, the primary control is exercised through the Symbol Engine, with the Neural Engine providing support to execute tasks. For example, Well et al. proposed training neural network classifiers to forecast viable motions and employing the classifier as a learned heuristic, steering the Task and Motion Planning (TAMP) search toward possible motions, thereby decreasing the total number of motion planning trials [83]. In [84], a Neural Engine, which takes the initial image of the environment as input, predicted the promising sequence of discrete actions, providing runtime improvements of several magnitudes. Given expert demonstrations, Chitnis et al. [85] applied learning techniques to efficiently search in the high-level task planning space, taking into account possible infeasibilities and, as a result, significantly increasing the planning speed of the Symbol Engine. Hiraoka et al. proposed a learning system that improves symbol grounding functions and a high-level planning method to optimize the total performance of the existing hierarchical planner in generating suitable plans [86]. Lamanna et al. [87] learned relational state representations, transition function, grounding function, and action-value functions to support the planning experience of a robot. PDDL planning was employed in the agent's exploration. Learning allowed the agent to scale up to larger environments. Next, they used the predicted confidence values from a Neural Engine to infer probabilistic belief states that were used by the Symbol Engine [88].

Dimensions: How the Sampled Studies Contribute
Control: Explicitly addressed in several: Anderson et al. [89] used piecewise linear functions as symbolic logic constraint policies, Cheng et al. [90] integrated task planning with RL, and Ahn et al. [76] trained low-level control policies, all supporting continuous action spaces; remaining studies used discrete actions.
Motion Planning: Addressed in several: motion parameters were predicted by GNN [82], and an SVM classifier invoked an RRT-based motion planner [83]; others used off-the-shelf motion planners [76], [90].
Safety: Integrated into RL exploration and learning via symbolic logic constraints [79], [89] and into an ontology-based failure detection framework [80]. Some studies addressed feasibility [82], [83] in terms of computational efficiency.
Architectures: Modular pipelines combining LLMs, planners, ontologies, FOL rules, and neural Q-functions [76], [79], [80].
Perception: Raw observations were processed by DNNs (ResNet) [88], neural visual encoders (Mask R-CNN) [78], and CNNs with geometric scene graphs [82]; neural grounding modules fed symbolic pipelines.
Applications: Long-horizon household manipulation [76], task-level instruction following [81], autonomous driving [79].

TABLE V
CONNECTION BETWEEN ROBOTICS AND THE STUDIES CATEGORIZED AS COUPLED (B).

B.3 Symbol Engines supporting Neural Engines

In this category, the Symbol Engine supports the Neural Engine, which serves as the main controller. Mota et al. employed a cascade of systems, where the domain knowledge encoded by a decision tree is compiled into an Answer Set Prolog (ASP) program that classifies images, and if unsuccessful, redirects the input to a convolutional network [91]. Ahmetoglu et al. used the principles of maximum information compression and slowly varying signals to extract symbol-like representations, enabling fast skill transfer [92]. The activations on the last hidden layers of the Neural Engines were used for this purpose. While the symbols enabled fast transfer, an explicit Symbol Engine was not fully utilized in this work.

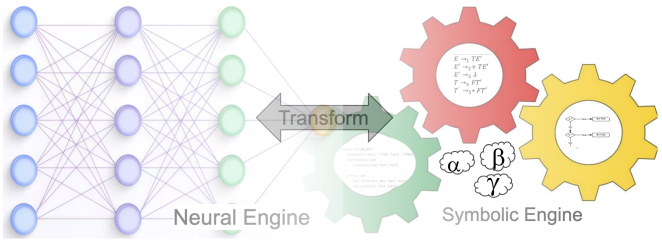


Fig. 5. The illustration of engine-to-engine transformation in the Neuro-Symbolic Translation (C) studies.

Cheng et al. leveraged the symbolic representation from the high-level planner to direct trial-and-error-based skill learning. Their system learns temporally extended actions to achieve the desired outcomes of the symbolic operators by using a reward, taking into account the post-condition of the operator within the Reinforcement Learning loop [90]. Rakhman et al. also proposed a method that utilized the Symbol Engine to determine exploratory actions for training the Neural Engine in a simulated mobile robot [93]. Last but not least, Lamanna et al. proposed using the PDDL Symbol Engine to improve the neural perceptual capabilities of the agent by selecting which objects to observe, which properties to utilize, and when to terminate data gathering, thereby improving its ability to recognize new object properties [94]. We end this section by summarizing the robotic capabilities that are offered by coupled neuro-symbolic systems (B).

Contributions to Robotics

The contributions of the systems categorized under the Coupled Neuro-Symbolic Robotics (B) to various dimensions of robotics are presented in Table V, along with representative studies. Compared to Intertwined Neuro-Symbolic Robotics, many of these studies were explicitly designed to address continuous control, motion and task planning, safety, and robotic perception. They learned continuous control policies via RL [90] or from demonstrations [76]. They could generate continuous parameters [82]. They also combined symbolic planning with neural feasibility estimation (e.g., TAMP heuristics [83] or symbolic and geometric scene graphs [82]), creating a scalable and efficient planning architecture. Safety is a major strength of these studies, which are addressed through symbolic logic constraints, ontologies, and explicit recovery strategies [79], [89]. In fact, the sole aim of some studies in this category was to detect and recover from failure [80]. Advanced neural networks took images as input and fed their output to symbolic planners, enabling robust grounding. Finally, these studies possessed strong real-world validation, including long-horizon manipulation [76], autonomous driving [79], and instruction-following in open environments [76].

Discussion

The coupled neuro-robotics studies loosely integrate Neural and Symbol Engines, such that the engines benefit from each other but are not forced to operate with each other. In many of these studies, multi-step planning is achieved through PDDL-like languages. Most of these studies use simulations for learning. They primarily verify their systems on real manipulator robotic platforms. Balanced neuro-symbolic robotics studies (B.1) benefit from the outputs of LLMs and, therefore, address natural language and communication. In the case of Neural Engines supporting Symbol Engines (B.2), the studies discuss the completeness, consistency, integrity, and correctness of their (symbolic) algorithms, emphasizing their verifiability. They also utilize the information learned by the Neural Engines for monitoring and fault detection. Therefore, B.2 category approaches might be used for safety-critical robot deployments. Symbol Engines supporting Neural Engines (B.3), on the other hand, are generally designed to guide the learning process of Neural Engines to increase the learning speed and/or quality through actively selecting actions and learning targets.

VI. C. Neuro-Symbolic Transformation

C.1 Transform Neural Engine to Symbol Engine

The robots controlled by Neural Engines generally lack explainability, interpretability, and verifiability, as we discussed in the Introduction section. To address this problem, some studies transformed the policies encoded by Neural Engines into symbolic representations (Fig. 5). For example, Verma et al. proposed an algorithm for learning a range of comprehensible skills with their parametric representations derived from the planning strategies encoded in the internal workings of the black-box AI agent [95]. For explainability and verifiability, Bastani et al. trained verifiable policies encoded with decision trees and demonstrated their framework in the cart-pole task within an RL setting [96]. Given a trained RNN, Weiss et al. learned an abstraction and extracted a deterministic finite automaton that encodes the state dynamics of the task [97]. For transparency, trust, explainability, and interoperability, Kaadoud et al. applied clustering in the latent space of the Long-Short Term Memory (LSTM) network, activated by each input sequence [98]. Using the hidden states, they generated a finite-state automaton that captured the underlying grammar, enabling the prediction of whether a given pattern is valid or not. To ensure compliance with behavioral specifications through formal guarantees, such as safety and/or reachability, Carr et al. proposed a method to autonomously build a finite-state machine from a recurrent neural network [99], thereby accommodating existing formal verification tools in agent benchmarks. These together indicate that the C.1 category approach can be used for safety-critical robotic applications.

C.2 Transform Symbol Engine to Neural Engine

In this category (Fig. 5), the manually encoded symbolic policies are transformed into continuous neural policies and refined through the robot's experience. Sun and Shoukry incorporated a Symbol Engine, which utilized linear temporal logic (LTL), into the training of a Neural Engine, such that each neural network in the system corresponded to a particular symbolic representation [100]. The resulting Neural Network-based planner inherited the symbolic model's interpretability and correctness assurances, with the aim of generalization

Dimensions: How the Sampled Studies Contribute
Control: Addressed via neural networks representing piece-wise affine functions to compute velocity [100], and linear regressors extracting decision policies to verify controller stability [96]; other studies focus discrete actions [97]–[99].
Motion Planning: Not studied extensively. [100] specifically designed a neuro-symbolic framework for motion planning with temporal goals; Carr et al. [99] extracted finite-state controllers and synthesized policies for temporal planning.
Safety: Integrated into learning/training loops for safe exploration [89], [100]. Carr et al. and Bastani et al. used finite-state controllers and decision trees to prove properties that are difficult to verify in neural networks [96], [99].
Architectures: Pipelines converting neural hidden states into symbolic structures [97]–[99] and back into neural policies [89], [100], enabling bidirectional transformation.
Perception: Not the primary focus; high-level discrete or continuous states are assumed as given [89], [96], [99].
Applications: Autonomous driving [89], [99], navigation [89], [99], walking and stability [96], mostly on datasets.

TABLE VI
CONNECTION BETWEEN ROBOTICS AND STUDIES CATEGORIZED AS NEURO-SYMBOLIC (C).

to unseen tasks, including new workspaces, novel temporal logic formulas, and errors in the robot's dynamical model. In [89], a Symbol Engine was used to implement symbolic policies, and another Symbol Engine was employed for formal verification of safety in every exploration loop of an RL-based robot learner. In their work, verifiable safe symbolic policies were transformed into continuous policies realized by the Neural Engine, which were updated using reward-based gradient learning and then transformed back to the original symbolic space. We expect an increase in the adoption of methods that transform one engine into another for more robotic tasks, ensuring verifiability and interoperability. Since this category allows the interpretation of the neural engine outputs in symbolic terms, they can be used in robot control with safety constraints. We conclude this section by summarizing the robotic contributions provided by the Neuro-Symbolic Translation methods (C.1 and C.2).

Contributions to Robotics

The contributions of the systems categorized under C to various dimensions of robotics are presented in Table VI, together with representative studies. In these studies, it was possible to provide continuous control signals to the robot by representing continuous functions with neural networks or extracting them from neural policies. While motion planning is not addressed in the majority of these studies, the transformations into symbolic structures allow planners to operate under symbolic constraints. This, importantly, enables formal verification (reachability, LTL satisfaction [100]) when the neural policies are symbolically represented. Perception for control or planning is typically abstracted away when extracting symbolic models, which makes these systems scale up to complex environments. Finally, although it has been demonstrated primarily on benchmarks and controlled tasks, the approach holds significant potential for safety-critical real-world domains that require verifiable behavior.

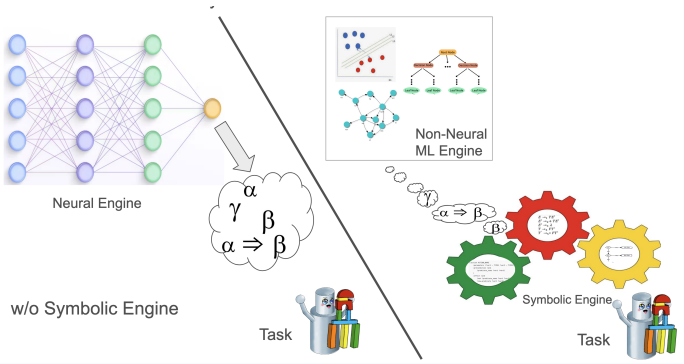


Fig. 6. Nonuniform Neuro-Symbolic Robotics (D).

Discussion

The studies we reviewed in this section share a common aim: to generate explainable and verifiable systems. These approaches do not explicitly use PDDL-like languages. As such, they do not benefit from LLMs. The systems that translate symbolic policies into neural policies (C.2) often lose their multi-step planning capabilities and are rarely implemented in robots. The neural-to-symbol transformation systems (C.1), on the other hand, can learn from RL, learning from demonstration or environment interactions, and gain planning capability, implemented mostly in manipulation platforms; yet their stronger safety guarantees come at the cost of scalability to high-dimensional perceptual inputs, as the abstraction required during neural-to-symbolic transformation limits their applicability to settings where rich sensory representations have already been reduced to lower-dimensional state variables.

D. NONUNIFORM NEURO-SYMBOLIC ROBOTICS

The studies in this category use both continuous and symbolic representations. However, the full-fledged power of one of the Symbol or Neural Engines is missing (Fig. 6).

D.1 Without Symbol Engine

In this category, Neural Engines are used to process (as input and output) symbolic and continuous representations to understand given tasks in natural language and for robot control. However, these symbolic representations are not exploited by the Symbol Engines.

LLMs bootstrap their models using pre-trained foundation models and leverage the multi-modal experiences of multiple robots operating in different environments to learn robot controllers capable of executing plans generated by the LLMs. To this end, large-scale data has been collected, like other AI systems that lack embodiment, despite the fact that collecting embodied experience for robots incurs significantly higher costs. Pioneering works in this domain include RT-1 [101], RT-2 [102], and RT-X [103]. The foundation models for robots, which integrate vision, language, and action, are referred to as vision-language-action (VLA) models [104], [105]. Given different goals and visual scene descriptions, these LLM-based systems can both generate a chain of actions and robot control commands, such as the target displacement of the robot's gripper at each step. Pre-trained VLMs were

utilized by exploiting semantic and syntactic relationships, disentangling action and perception, and producing control parameters for given manipulation primitives [106]. Pallagani et al. implemented a transformer-based Neural Engine that takes the problem and domain in symbolic representation (in PDDL) as input and generates the sequence of actions to solve the problem [107]. These studies are included in this survey because the Neural Engine can take goals represented as symbols and generate intermediate steps in symbolic form. On the other hand, they do not benefit from Symbol Engines; therefore, the generated plans are neither directly explainable nor verifiable.

Namasivayam et al. [108] extended [109] to the robotics domain and proposed a "Neuro-Symbolic program" that processes both continuous and symbolic representations using trained Neural Engines. It trains a Neural Engine to parse natural language instructions, transforming them into a program in Domain-Specific Language (DSL). Other Neural Engines, such as a Visual Extractor, receive the visual scene and produce visual features. The DSL description and visual features are combined as inputs in the Neural Engine Visual Reasoner that outputs an action. The framework is trained end-to-end. In [110], to discover rules from demonstrations, a graph neural network (GNN) was learned, where nodes encode task and domain-related entities, such as objects and outcomes. Long-horizon planning was performed using a gradient-based heuristic, which does not use symbolic knowledge. Interestingly, to add interpretability, they determined the importance of neighboring nodes in decision-making and allowed explanations such as "this node was selected because of its connection with this and these nodes; the most relevant feature being this particular object feature". Zadem et al. proposed a developmental progression for symbolic sub-goal discovery in a hierarchical RL framework that combines states with similarities for the given tasks [111]. They further extended this work to learn both spatial and temporal goal symbols [112], [113]. Focusing on the reachability problem in a mobile robot, a high-level agent finds regions in the reachability-aware goal space, and other agents select the sub-goal symbols for reaching goals and learn how to execute the corresponding actions, increasing the learning speed and scalability. No Symbol Engine is used in these approaches.

Beyond the planning-oriented systems above, a number of studies incorporate symbolic knowledge in the physics layer: as analytical rigid-body dynamics and contact laws in NeuralSim [114], as an explicit symbolic prior, for the analytical quadrotor ODE, embedded inside a physics-informed neural network loss in RAMP-Net [115], and as a 6-DOF analytical dynamics framework within which a neural network approximates unknown pitch dynamics in the AUV controller [116]. In all three cases, however, symbols structure the physics without driving any planning or reasoning layer, which is the defining characteristic of D.1.

Contributions to Robotics

The contributions of the systems categorized under D.1 to various dimensions of robotics are presented in Table VII, together with representative studies. These studies utilized powerful neural network models that are capable of processing visual input, grounding symbols, and generating low-level action signals, high-level action parameters, or action trajectories. Planning and reasoning were achieved through VLMs/LLMs rather than explicit symbolic engines. Safety is the weakest across all categories, with minimal symbolic validation and high reliance on neural generalization, making failures unpredictable in long-horizon tasks. Because LLMs are prone to hallucination, these studies have attempted to address safety issues at the low-level motion planning and control level, rather than in reasoning or decision-making. The models, such as RT-1 and RT-2 [101], [102], achieved strong practical control in real settings; however, the lack of symbolic oversight means these systems cannot guarantee consistency or correctness in edge cases. They excel in perceptual generalization, leveraging large-scale multimodal datasets (RT-X [103]) producing rich representations. Overall, these studies have the broadest real-world impact, powering robot foundation models and large-scale instruction-following robots, but lack the transparency and safety guarantees required for real-world deployment.

Dimensions: How the Sampled Studies Contribute
Control: Continuous control signals (actuator torques, manipulation targets) were generated via RL [112] and predictive neural networks [104], [108]; other studies discretized action spaces into bins [101]–[103].
Motion Planning: Not addressed except in [104], where future trajectory predictions were used for motion planning.
Safety: Delegated to external modules in most studies [108], [110], [112]. An exception is [104], which explicitly prompts for potential risks and safety-critical factors.
Architectures: End-to-end neural pipelines combining DSL parsing, visual reasoning, and action generation [108]; robot foundation models integrating vision, language, and action [101]–[103].
Perception: Perception from raw visual data for control and symbol grounding was performed by pre-trained vision encoders [50], [101], [102], [104], [108].
Applications: Large-scale manipulation [102], multi-robots [103], mobile manipulation [113], instruction following [108].

TABLE VII
CONNECTION BETWEEN ROBOTICS AND THE STUDIES CATEGORIZED AS NONUNIFORM - W/O SYMBOL ENGINE (D.1).

D.2 With Non-Neural ML

Here, we review studies that do not explicitly use generic Neural Engines but benefit from various Machine Learning techniques to learn/process symbols, making inferences and plans with Symbol Engines. They generally have not utilized neural networks and historically appeared earlier than the studies we have reviewed so far. Still, we would like to include these studies as they paved the way for Neuro-Symbolic Robotics. We review them using the same convention as we have for our neural counterparts above.

Initial studies addressed learning sub-symbolic structures that were useful in planning. In the seminal work of [117], the interaction experience of a mobile robot was used to cluster low-level sensory data into categories through self-organizing maps. The system made plans by predicting the

next sensory state, where each state was represented by one of the found clusters. Similarly, Ugur et al. applied clustering in the effect space in manipulation and mobile manipulation domains [118], [119]. Their system operates by first identifying effect categories and then learning SVM classifiers that map initial features to effect categories, effectively forming action-effect predictors used for planning. Mugan and Kuipers learned discrete representations from environmental features and a set of predictive models (Dynamic Bayesian Networks) based on these discrete symbols, generating symbolic plans and executing them in a simulated robot to manipulate blocks [120]. In these studies, symbols were discovered through unsupervised robot interaction, similar to the studies in category A.2 (Symbol Discovery), but without Neural Engines.

The studies above address the symbol emergence problem using a combination of clustering and discriminative ML techniques. On the other hand, we mentioned that probabilistic generative models can also be used to learn symbols. A series of studies following MLDA allowed robots to form variable-number [121], hierarchical [122], and modality-weighted [123], [124] multi-modal object categories [34]. Furthermore, Araki et al. introduced an unsupervised word segmentation method capable of discovering words in an unsupervised manner alongside MLDA-based multi-modal object categorization [125]. Nakamura et al. [126] and Nishihara et al. [127] subsequently integrated word segmentation and categorization into a single probabilistic model, enabling a robot to discover words and categories entirely in an unsupervised manner. More recently, Kuniyasu et al. explored object category discovery and multi-modal symbol learning using Modal Latent Dirichlet Allocation (MLDA) and variational autoencoders [128]. Similar approaches have been applied to spatial concepts by integrating Simultaneous Localization and Mapping (SLAM), Gaussian Mixture Models (GMMs), speech recognition, and image recognition models [129]–[132]. These studies enabled robots to autonomously acquire spatial concepts, relative spatial concepts, and language, demonstrating an unsupervised pathway from multi-modal category formation to language acquisition. Last but not least, Hasegawa et al. [133] presented an early example of a neuro-symbolic approach integrating spatial concepts with probabilistic logic.

Konidaris et al. [29], [140], [141] discussed that high-level planning can be achieved by learning discrete symbols used to encode the preconditions and the postconditions of an agent's action repertoire. They learned symbols to encode action preconditions and effects, and used them in the operators for building a PDDL description of the environment. Ames et al. learned symbolic operators along with skill parametrization in [139]. By pre-processing an incoming image with independent component analysis followed by image-to-symbol mapping via Support Vector Machines, Pacheck et al. [135] proposed a method to form symbols from raw images, following [29]. Then, the given robot skills were encoded in Linear Temporal Logic (LTL), enabling symbolic planning for tasks written using LTL formulas. James et al. [142] extended this approach by using an agent-centric as opposed to object-centric encoding, allowing the discovered symbols to be transferred across different environments. In follow-up work, they demonstrated that the generalization capability can be enhanced through the use of object-centric representations [143]. Ugur et al. also learned object-centric symbols used in preconditions and effects of PDDL operators [136], [137]. Typically, the first step in these studies involved applying clustering to the observed interaction instances to find a discrete representation. As such, the quality of the learned symbols relied on the quality of the state-space partitioning, which was an unsupervised process with no guarantees on the latter planning performance.

An RL-based symbol learning framework was proposed in [138], where learned symbolic relational abstractions are used to encode transition and reward models, predicting action effects, and ultimately, multi-step planning. The optimization of symbol learning focuses not only on enhancing effect prediction performance but also on maximizing rewards. Finally, Rosen et al. [134] learned preconditions of manipulation and navigation operators by leveraging the distinction between spatial and non-spatial state variables. The independence assumption between manipulation and navigation operators enables planning using only manipulation skills, and then automatically filling in the navigation steps.

While the above studies focus on single-robot manipulation and navigation, Jung and Zelinsky [144] and Paxton et al. [145] apply this non-neural symbolic approach to multi-robot and collaborative settings: the former coordinated heterogeneous robots through hand-engineered iconic and indexical groundings, while the latter composed non-neural object- and pose-level perceptual symbols into Behavior Tree task plans for collaborative manipulation, both illustrating the grounding and planning pipelines that neuro-symbolic approaches aim to learn automatically.

Dimensions: How the Sampled Studies Contribute
Control: Rarely addressed; motor actions are sampled from motor babbling in [120], while the majority assume manually coded action primitives.
Motion Planning: Addressed in a few studies: an external path planning algorithm in [134] and a heuristic propagation mechanism in action links in [117].
Safety: Not explicitly addressed except in [135], where Linear Temporal Logic encodes the robot's safety constraints.
Perception: Perceptual grounding used traditional methods: dimensionality reduction via PCA/ICA [29], [135], feature extraction via histograms of surface normals [118], [119], and classification via SVMs [135], [136].
Applications: Table-top manipulation [118], [135]–[138], mobile manipulation [29], [119], [139], 2D navigation [140], [141].

TABLE VIII
CONNECTION BETWEEN ROBOTICS AND THE STUDIES CATEGORIZED AS NONUNIFORM - W/ NON-NEURAL ML (D.2).

Contributions to Robotics

The contributions of the systems categorized under D.2 to various dimensions of robotics are presented in Table VIII, together with representative samples. These studies, in general, assume pre-defined action primitives and focus on learning action possibilities provided by the environment and symbolic representation of these possibilities. As such, motion planning or safety concerns are not included in their

learning loop with a few exceptions. As these works use traditional feature extraction and machine learning techniques for perceptual processing, they can be viewed as the precursors of neuro-symbolic approaches. Generally, constrained by the computational resources of the era, their applications were limited to relatively simple manipulation and navigation tasks in simulation and real-world settings.

Discussion

In this section, we reviewed a mixed body of literature covering architectures that do not explicitly use either symbolic or neural engines, yet are strongly related to neuro-symbolic learning. Common to the Symbolic robotic systems with non-neural ML (D.2) predate their neural counterparts and do not provide explainability, verifiability, and execution monitoring. However, they are verified in a wide range of simulated and real robotic systems, including mobile and manipulation robots. Except in a specific research direction, where probabilistic graphical models discover perceptual symbols related to language symbols, language communication is not at the core of these studies. The studies that do not include a symbol engine (D.1), on the other hand, utilize LLMs for multi-step planning, primarily learn from human demonstrations, and do not employ active learning or intrinsic motivation. They also rarely discuss explainability or verifiability.

VII. General Discussion

A. Neuro-Symbolic Robotics in the Age of LLMs

Recent advances in LLMs have significantly blurred the traditional distinction between symbolic reasoning and continuous/sub-symbolic neural processing. While symbolic AI and symbol engines have long been valued for their structured and verifiable logical inference, LLMs have increasingly demonstrated the ability to perform step-by-step reasoning through techniques such as Chain-of-Thought (CoT) [146], and Tree-of-Thought (ToT) [147]. Yet this reasoning is not accompanied by formal verification: LLM usage reaches 100% in A.3, 86% in B.1, and 73% in D.1, while verifiability in these same categories is 0% in both B.1 and D.1, and only modest in A.3. This numerical disconnect directly supports the claim that LLM-based reasoning, however impressive, does not constitute formally verifiable inference. Additionally, neuro-symbolic approaches such as ReAct [148] and ToolFormer [149] enhance inference accuracy by integrating external symbolic systems as reasoning tools. Given that these symbolic tools function as external programs, this development underscores a convergence between neural and symbolic methods, reinforcing the role of neuro-symbolic architectures in robotic cognition.

One should, however, note the distinction between a formal symbolic system and an external symbolic system such as language. The properties of provability, explainability, and disambiguity pertain only to the former. LLMs demonstrated that non-trivial computation can be undertaken even without such guarantees, provided we are willing to forgive some mistakes. LLMs model external symbolic structures, such as natural language, which are inherently ambiguous and contextual yet flexible and open to change. In contrast, formal symbolic AI captures mathematical logic, thus has a rigid structure that is free from cultural evolution. Although it may be tempting to argue that human cognition generates language but computes symbolically internally, the average human seems to compute with language as exemplified by the human fallacy of interpreting A implies B as equivalent to double implication during conversations [150]. Even without this human paradox, it remains an unsettled issue whether any symbolic system underlies neural computation, and if so, what kind of symbols it uses [4]. Yet, the effectiveness of LLMs in reasoning suggests that maintaining a continuous internal representation while leveraging external symbolic reasoning can be a clever way to harness the benefits of LLMs and formal symbolic systems for more capable artificial systems.

In the context of Neuro-Symbolic Robotics, the distinction between what becomes internal and what becomes external is crucial for understanding how symbolic representations emerge and function within artificial multi-agent systems. On the one hand, LLMs utilize formal symbols as external tools to enhance their performance. On the other hand, some robotic studies on symbol emergence demonstrate the expansion of internal representation formation into external representation sharing, resembling models of emergent communication and language evolution [151]–[154]. Future research should explore ways to align the workings of LLMs with an underlying internal formal system, paving the way toward trustworthy and explainable AI systems that can scale.

B. Implications for Robotics

In this paper, we examined the potential contributions of individual Neuro-Symbolic Robotics categories to robotics across five dimensions: robot control, motion planning, safety, robot perception, and robotic applications (see Tables II–VIII). Table IX provides a qualitative summary of the presence of key aspects across all reviewed categories, where '-' indicates the aspect is addressed in fewer than 20% of sampled studies, '✓' indicates it is addressed in $20-60\%$ of studies, and '✓✓' indicates it is addressed in more than 60% of studies. This table is discussed quantitatively in Section V.B, where we extract column-wise implications for each robotics dimension. In this section, we adopt a complementary, dimension-centric (column-wise) perspective, summarizing the overall implications of Neuro-Symbolic Robotics as a whole for each robotics dimension.

Robot Control: The core modules of Intertwined Neuro-Symbolic Robotics (A) studies generally do not explicitly address low-level control. Still, the learned high-level affordance detectors and outcome predictors can be used before and during action execution for control. Coupled Neuro-Symbolic Robotics (B) approaches provide the most robust control capacity, where independently trained neural policies are generally used to control the robot's execution while being guided or monitored by high-level symbolic input. Note that the control provided by the foundation models might generalize well to different environments but may fail to comply with task constraints. There are a few studies in Neuro-Symbolic Translation (C) where neural network-generated parameters

TABLE IX
DIFFERENT ASPECTS STUDIED IN NEURO-SYMBOLIC ROBOTICS FOR EACH CATEGORY. THESE ASPECTS ARE ADDRESSED ONLY RARELY ('-'), BY A FEW STUDIES ('✓'), AND BY MOST STUDIES '✓✓.' PLAN (PERFORMED MULTI-STEP PLANNING), PDDL (USED ADL OR PDDL), LLM (USED LLMS), LANG (USED IN THE LANGUAGE), EXPL (AIMED EXPLAINABILITY), VER (EMPHASIZED VERIFIABILITY), MON (USED IN MONITORING OR FAULT DETECTION), PRED (USED FOR NEXT STATE PREDICTION), RL, LFD , RAND (RANDOM EXPLORATION), ACTL (ACTIVE LEARNING OR INTRINSIC MOTIVATION), INTER (LEARNED FROM INTERACTIONS), REAL (REAL ROBOTS), SIM (SIMULATED ROBOTS), MBL (MOBILE ROBOTS), MNP (MANIPULATORS).

Category	Plan	PDDL	LLM	Lang	Expl	Ver	Mon	Pred	RL	LfD	Rand	ActL	Inter	Real	Sim	Mbl	Mnp
A.1.a	✓✓	✓✓	-	-	✓	-	✓✓	✓✓	-	✓	-	-	✓✓	✓✓	✓✓	-	✓✓
A.1.b	✓✓	✓	-	-	-	-	✓	✓✓	✓	✓	-	-	✓✓	✓✓	✓✓	✓	✓✓
A.2.a	✓✓	✓	-	✓	-	-	-	✓✓	✓	-	✓	-	✓✓	✓	✓✓	-	✓✓
A.2.b	✓✓	✓✓	-	-	-	-	-	✓	✓	✓	✓	✓	✓	✓	✓✓	✓	✓✓
A.2.c	✓✓	✓✓	✓	-	-	-	-	-	-	✓	-	-	-	✓	✓✓	✓	✓✓
A.3	✓✓	✓✓	✓✓	✓	✓✓	✓	✓	✓✓	✓	✓✓	✓	-	✓✓	✓✓	✓✓	✓✓	✓✓
B.1	✓✓	✓	✓✓	✓✓	✓	-	✓	-	✓	✓	✓	-	✓	✓	✓✓	✓	✓✓
B.2	✓✓	✓✓	-	-	-	✓✓	✓✓	✓✓	✓	✓✓	✓✓	✓✓	✓✓	✓	✓✓	✓	✓✓
B.3	✓✓	✓✓	-	✓	✓	✓	✓	✓✓	✓	✓	✓	✓	✓✓	✓	✓✓	✓	✓
C.1	✓✓	-	-	-	✓✓	-	-	✓	✓✓	✓	-	-	✓	-	✓✓	-	✓
C.2	-	-	-	-	✓✓	✓	-	-	-	-	-	-	✓	-	✓	-	-
D.1	✓✓	-	✓✓	✓✓	✓	✓	-	✓✓	✓	✓✓	✓	-	-	✓✓	✓✓	✓	✓✓
D.2	✓✓	✓✓	-	✓	✓	✓✓	-	✓✓	✓	-	✓✓	-	✓✓	✓✓	✓✓	✓✓	✓✓

are verified for stability, thereby contributing to controller robustness.

Motion Planning: The core modules developed in Intertwined Neuro-Symbolic Robotics (A)do not explicitly address motion planning. By contrast, Coupled Neuro-Symbolic Robotics (B) approaches offer strong geometry-aware motion planning, combining symbolic reasoning with neural feasibility estimators and scene graphs, thereby enabling the integration of task and motion planning. Neuro-Symbolic Translation (C) methods focus on verifiable symbolic abstractions but lose geometric detail, which means they cannot inherently support low-level motion planning for execution.

Safety: Safety is not directly and strongly addressed in studies except in Neuro-Symbolic Translation (C), which integrates safety into its learning or training loop. These studies lead to formal verification, enabling reachability, correctness, and temporal logic guarantees through symbolic abstraction of neural models. The symbols, rules, and programs learned or discovered in Intertwined Neuro-Symbolic Robotic (A) studies and the sensorimotor trajectory models generated in Coupled Neuro-Symbolic Robotic (B) studies can be used for motion and task monitoring for safe task execution. However, in general, safety aspects are not the main focus of these studies. Quantitatively, safety is explicitly addressed in only 21% of all reviewed studies. Across all A.2 subcategories, which collectively represent the largest group of symbol-discovery studies, safety scores are 0%, confirming that neural-dominant symbol discovery approaches systematically neglect these properties.

Robot Perception: In Intertwined Neuro-Symbolic Robotics (A), perception is usually task-specific especially in studies where symbols are learned (category A.1), whereas it is more domain-driven in studies where symbols are discovered (A.2). Coupled Neuro-Symbolic Robotics (B) integrates neural perception with symbolic reasoning (e.g., scene graphs and visual grounding), achieving perceptual grounding for downstream tasks. In contrast to the aforementioned approaches, in Neuro-Symbolic Translation (category C) studies, perceptual processes are not the primary focus, and as such, they assume that high-level features are directly available.

Robotic applications: Real-world deployment capability of Intertwined Neuro-symbolic approaches (A) depends on whether perception is modeled or assumed. In studies where symbols are learned or discovered (A.1 and A.2), performance in dynamic or unstructured domains is limited, although their excellent performance in structured environments. On the other hand, the studies in which symbolic programs are learned by neural networks (A.3) exhibit growing real-world potential. Neuro-Symbolic Translation (C) approaches also hold potential for real-world deployment as they emphasize safety and verification. B.2 leads all categories on safety (75%) and verifiability (83%), B.1 achieves the highest LLM integration (86%) with 57% real-robot deployment, and across all B sub-categories multi-step planning reaches 100% while explicit continuous control is more prevalent here than in any other category. No other category combines this breadth of robotics-relevant dimensions simultaneously. Overall, Coupled Neuro-Symbolic Robotics (B) currently represents the most capable paradigm for addressing the breadth of real-world robotic challenges, with demonstrated applications in household manipulation, autonomous driving, instruction following, and long-horizon mobile manipulation tasks.

The key observations directly extractable from Table IX are as follows: **Planning** is near-universal across all categories (82–100%), confirming it as the dominant motivation in neuro-symbolic robotics, with the sole exception of C.2 (0%), where symbolic-to-neural translation sacrifices planning capability entirely. **Verifiability** shows a striking contrast: it is addressed in 83% of B.2 studies and 50–83% of C.1 and C.2 studies, but drops to 0% across all A.2 subcategories and B.1, highlighting a critical gap between developmental symbol discovery approaches and safety-oriented engineering approaches. **LLM usage** is highly concentrated: 100% in A.3, 86% in B.1, and 82% in D.1, but 0% across all other categories, confirming that LLM-based neuro-symbolic robotics is a recent and distinct sub-trend rather than a field-wide phenomenon. **Real robot deployment** is highest in A.1 (86–100%), A.3 (75%), and D.1–D.2 (73–78%), but drops to 0% in C.1 and C.2, revealing that the Neuro-Symbolic Translation category, despite its strong safety and verifiability credentials, remains almost

TABLE X
CROSS-CUTTING QUANTITATIVE SUMMARY OF REVIEWED STUDIES. REAL = REAL ROBOT; SAFE = SAFETY; PLAN = MULTI-STEP PLANNING.

Cat.	N	Real (%)	Safe (%)	LLM (%)	Plan (%)
A.1.a	3	67	0	0	100
A.1.b	6	71	14	0	100
A.2.a	11	33	0	0	100
A.2.b	12	45	0	9	100
A.2.c	7	50	0	25	100
A.3	4	80	40	100	100
B.1	6	57	29	86	100
B.2	6	38	75	0	100
B.3	5	33	33	0	100
C.1	5	0	50	0	83
C.2	2	0	50	0	0
D.1	15	73	18	73	82
D.2	30	78	6	0	89
Total	**112**	**54**	**21**	**27**	**96**

entirely unvalidated on physical robots. C.2 additionally drops to 0% on multi-step planning, the only category to do so, numerically capturing the precise trade-off of this paradigm: strong on verification, absent from the physical world and from long-horizon task execution. **Safety** (Ver + Mon combined) is most strongly represented in B.2 (83% verifiability, 83% monitoring) and C.1/C.2, but is near-absent in A.2 (0% across all subcategories), despite A.2 being the largest group of studies, representing a critical gap between symbol discovery research and safe deployment requirements. D.1, despite being one of the larger categories and achieving the highest LLM usage at 73%, scores only 18% on safety (contrasting sharply with B.2 at 75% and C.1 at 50%), confirming that removing the symbolic engine comes at a direct and measurable cost to safety coverage. **Mobile robotics** (Mbl) is underrepresented across all categories, reaching 75% only in A.3 and dropping to 0% in A.1.a, A.2.a, C.1, and C.2, giving quantitative support to the manuscript's argument that mobile robotics remains under-served by neuro-symbolic approaches.

Table X provides a complementary cross-cutting quantitative summary across the entire reviewed literature:

- **Robot Control:** Explicit continuous control is addressed in only 38% of all reviewed studies, with Coupled Neuro-Symbolic approaches (B) accounting for the majority of these cases.
- **Safety:** Safety is explicitly addressed as a primary design concern in only 21% of all reviewed studies. It is most prevalent in B.2 (75%) and C.1 (50%) categories, and virtually absent in A.2 (0%) and D.2 (6%) categories, highlighting a critical gap in the field.
- **LLM integration:** LLMs are used in 27% of all reviewed studies, concentrated almost entirely in A.3, B.1, and D.1 categories, confirming that LLM-based neuro-symbolic robotics is a recent and rapidly growing sub-trend.
- **Real-world deployment:** Real robot experiments are reported in 54% of all reviewed studies. Note that this real-to-simulation ratio approximately holds across platforms with the exception of autonomous driving as shown in Fig. 8. This figure drops to 0% in the Neuro-Symbolic Translation category (C), which has been validated almost exclusively in simulation and benchmarks, underscoring the gap between its theoretical promise and practical deployment.
- **Multi-step planning:** Multi-step planning is nearly universal across the reviewed literature (96%), confirming that planning is the primary motivating application for neuro-symbolic robotics research.

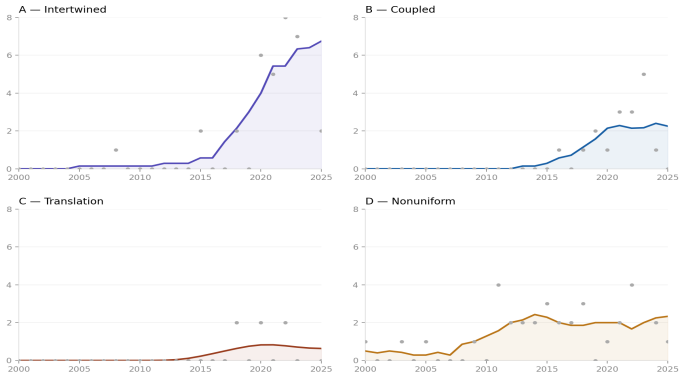


Fig. 7. Temporal distribution of reviewed studies per category (2000–2025). Curves show moving average of annual study counts; gray dots indicate individual data points. Outliers were removed via IQR prior to smoothing.

Finally, Fig. 7 complements the Table X by placing these category-level counts in a temporal context: while D has the longest continuous presence owing to its non-neural precursors, A and B exhibit sharp growth after 2018, and C remains consistently sparse between 2015-2025, reflecting the narrow yet technically demanding nature of translation-based approaches.

A few recurring themes stand out from these comparisons. First, formal verifiability and real-world adaptability seem to pull in opposite directions: Category C approaches have the strongest safety credentials but poor real-robot deployment, while Category A shows exactly the opposite. Grounding formally verified abstractions in autonomously discovered symbols, is an important unsolved problem in this regards. Second, symbolic bottlenecks come up as a reasonable middle ground, since they improve interpretability without fully giving up neural flexibility, though whether they can be made both formally verifiable is still an open question. Third, there is a large gap between formally verified planners (Category C) and end-to-end vision-language-action systems (Category D.1). The former gives correctness guarantees but struggles with perceptual grounding, while the latter handles perception well but tends to break down in edge cases. Building systems that can autonomously discover verifiable abstractions straight from sensorimotor experience seems like the most promising direction here.

C. Challenges

In the previous section, we discussed the contributions of the Neuro-Symbolic studies to robotics. Here, we reveal the missing components and important challenges in the field. **Multi-step planning** is achieved by all studies except those where symbolic policies are transformed into their neural counterparts, thereby losing the symbolic planning capability. While this transformation is done to refine the corresponding policies in the robot's continuous sensorimotor experience,

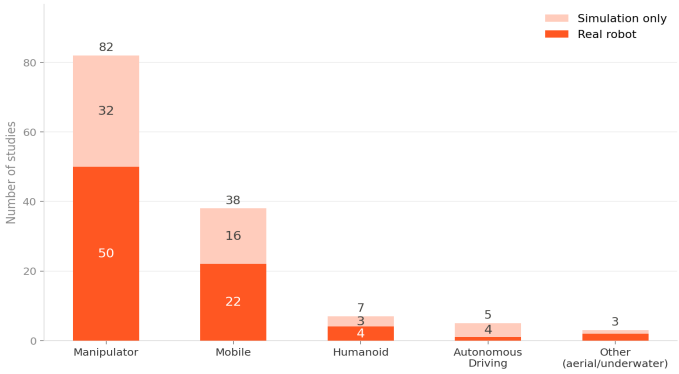


Fig. 8. Distribution of robotic platforms across the reviewed studies. Dark bars indicate studies validated on real robots; light bars indicate simulation-only evaluations. Studies may appear in more than one platform category.

this comes at the expense of losing many important cognitive capabilities. Therefore, one challenge is to refine symbolic policies in the robot's environment, retaining the high-level cognitive capacity of the system. **ADL and PDDL** are generally used as the main planning languages. However, they are not sufficiently addressed as part of the translation of neuro-symbolic transformations. We see a wide opportunity to use LLMs to translate the outputs of neuro-symbolic processing into the PDDL domain. **Natural language processing and communication** in neuro-symbolic robotics have been addressed in studies where LLMs play a major role. This is a natural tendency. However, there is still a gap between the symbols provided by LLMs and those generated autonomously through the robot's interactive exploration. Closing this gap is an important challenge that needs to be addressed; equally pressing are the energy and onboard deployment constraints that LLM-driven systems impose, as their substantial computational and memory demands, currently requiring cloud offloading or high-end hardware, conflict with the autonomy, latency, and power constraints of mobile and field robots.

Explainability, verifiability, and task monitoring are not strongly addressed by any of these systems simultaneously (no single category achieves '✓✓' (>60%) in these aspects simultaneously in Table IX), which affects their adoption in real-life and industry applications. Hence, robust research programs are necessary to meet the requirements for widespread adoption. As shown in Table IX, RL achieves ✓✓ in only 1 out of 13 subcategories, whereas LfD learning from demonstration and interaction-based learning reaches 3 and 7 ✓✓ subcategories, respectively. This data shows that neuro-symbolic robotics generally learns from demonstrations or the robot's own interaction experience, and **reinforcement learning** for learning neuro-symbolic structures has not received sufficient attention. Integrating reward-based trial and error, as well as learning from human and robot experiences, are important open challenges in neuro-symbolic robotics. Most of the literature on neuro-symbolic robotics is tested on manipulation tasks with robotic manipulators. Mobile robotics, on the other hand, has not been sufficiently addressed with neuro-symbolic approaches. Fig. 8 confirms this imbalance quantitatively: manipulator robots account for the vast majority of reviewed studies, with mobile platforms a distant second, and humanoid, autonomous driving, and aerial/underwater systems collectively representing only a small fraction of the literature. **Active learning and intrinsic motivation** are not exploited sufficiently for neuro-symbolic learning (achieved ✓ (20−60%) in only 3 out of 13 subcategories as shown in Table IX). This makes *active neuro-symbolic learning* a rich venue to study. Finally, although **multimodal approaches** exist, the richness of tactile manipulation and force-based skill learning to their full extent has not been addressed with the current neuro-symbolic studies. For example, can neuro-symbolic systems discover and utilize different modes of contact or force control modes for interacting with the world, such as when using a paintbrush, a pen, or a power drill? These should require different sets of symbols to be controlled effectively by high-level planning.

D. Future Research Directions

1) Safety-Critical Domains and Verification: Unlike disembodied Neuro-Symbolic AI, Neuro-Symbolic *Robotics* directly impacts the physical world, making safety and explainability paramount. As LLM-based inferences can suffer from hallucinations [7], deploying these systems requires rigorous verification mechanisms, especially in human-robot interaction scenarios. Future application domains should prioritize areas where safety is critical, such as collaborative robots in manufacturing and assistive robotics in healthcare, where explainability is not optional but mandatory. Additionally, extending these architectures beyond manipulation to mobile robotics, addressing long-term spatial concept learning and decision-making in complex environments, remains a vital frontier for validating the scalability of Neuro-Symbolic approaches.

2) Potential Neuro-Symbolic Approaches for Robotics: The full potential of advanced neuro-symbolic approaches has yet to be fully leveraged in robotics. For example, in [155], a Neural Engine generates predicates from continuous inputs such as images, where these predicates are used in their probabilistic logic programming framework. In [156], predicates are also realized as the outputs of neural networks over continuous features, and logical formulas are composed using fuzzy operators. Inference is performed via gradient optimization instead of complete logical inference. NeurASP [157] is an extension of Answer Set Programming (ASP) that treats Neural Engine outputs as probability distributions over ASP atoms, which can identified from continuous traces of labeled demonstrations [158]. These studies can be related to the Coupled Neuro-Symbolic Robotics in our taxonomy; however, their expressive inference power has not been utilized in robotics yet.

3) Grounding Symbols in Embodiment and Interaction: A fundamental challenge lies in bridging the gap between the top-down, linguistic symbols provided by LLMs or conventional symbolic planners and the bottom-up, internal symbols acquired through robot interaction (as reviewed in Section IV-A2). Current approaches are heavily biased towards vision; however, true embodiment requires grounding symbols in rich sensorimotor modalities, such as tactile and force information [7]. Theoretical frameworks such as collective predictive

coding (CPC) [159], which posit that external symbols emerge through distributed inference, offer a promising route to understanding how agents dynamically form symbols. This may lead to the utilization of symbol systems with granularities and semantics optimized for the robot's specific embodiment, potentially differing from human language but facilitating more effective human-robot collaboration.

4) Bridging the Gap between VLA and Neuro-Symbolic Planning: While VLA models [105] have demonstrated impressive capabilities in end-to-end control (discussed in Section IV-D), they currently lack the verifiability and long-horizon reliability required for complex tasks. A critical research direction is to integrate these foundation models with the structured reasoning of Neuro-Symbolic architectures. Beyond classical symbolic planning, emerging techniques such as CoT [146] and Code as Policies [160] offer new ways to utilize LLMs for planning. Furthermore, as seen in ToolFormer [149], integrating external symbolic tools can enhance inference accuracy. Developing frameworks that embed these reasoning capabilities into the robot's action generation loop, effectively moving from Nonuniform to Intertwined or Coupled systems, is essential for creating competitive solutions for industrial and domestic environments.

E. Revisiting Neuro-Symbolic AI

The Neuro-Symbolic AI field studies the integration of neural and symbolic systems without any explicit reference to robotic systems. In [161], Garcez et al. discuss neuro-symbolic AI from representation, learning, and reasoning, knowledge extraction, and transfer and consolidation aspects; and argue that structure learning, learning generalizations of symbolic rules, and knowledge extraction from large neural networks are key challenges in this field. In [5], Besold et al. provide a review of neuro-symbolic AI, where they discuss neural-symbolic computation around three core aspects, namely, principles, mechanisms, and applications of neural-symbolic integration. In addition, the authors discuss various approaches based on the types of integration, including logic systems and neural networks, logic, connectionism, and models of uncertainty, as well as approximation methods. A categorization for Neuro-Symbolic AI was proposed by Kautz et al., where neuro-symbolic AI approaches were grouped into six different categories [162]. The first two categories (Symbolic Neuro Symbolic approach and Symbolic[Neuro] architecture) are covered by the the neural part of our Coupled Neuro-Symbolic Robotics (B); their third category (Neuro—Symbolic system) corresponds to symbol learning part of Intertwined Neuro-Symbolic Robotics (A.1); their fourth category (Neuro: Symbolic $\rightarrow$ Neuro approach) corresponds to one direction in Neuro-Symbolic Translation (C.1) and finally the last categories are not realized in robotics. In short, while existing neuro-symbolic AI surveys provide valuable insights, they briefly mention robotic studies whereas we provide a critical review of the neuro-symbolic robotic studies and provide a more comprehensive and detailed taxonomy that is designed not only based on neural-symbolic integration mechanisms but also driven by the the embodiment and interaction dynamics of the robot with the environment (discriminating the grounding, pre-conditions and effects of actions), sensorimotor capabilities of the robots (discriminating perceptual, action and percept-action symbols), and the goal of the robots (discriminating effect prediction and planning).

VIII. Conclusion

In this paper, we reviewed the studies in the recently emerging field of Neuro-Symbolic Robotics. We proposed a taxonomy of these studies, categorizing them based on the roles of the Neural and Symbol Engines in the respective robotic architectures and how these engines interact with each other. While there has been significant effort in non-robotics neuro-symbolic agent architectures [1], [5], [6], [8], [9], [161], [163], we concluded that its robotics counterpart is in its initial stages. We also note that while each category addresses different challenges. These challenges include how a Neural Engine can discover symbols useful for the Symbol Engine, how Neural and Symbol Engine outputs can be combined for intelligent and robust control, or how Neural Engines can be transformed into Symbol Engines for verifiability and interoperability. However, an integrated robotic architecture that fully utilizes the benefits of both neural and symbol systems has yet to be introduced. This is not merely a qualitative observation: across the 13 subcategories reviewed, no single category simultaneously achieves high scores across all key dimensions. D.2 leads on real-robot deployment (78%) but scores only 6% on safety; B.2 leads on safety (75%) but scores 0% on LLM use; C.1 and C.2 score well on safety but 0% on real-robot deployment. This fragmentation, with no category dominating all dimensions, numerically illustrates that a fully integrated solution remains out of reach. LLMs, especially VLMs, have been gaining great popularity in neuro-symbolic robotics as they can analyze low-level sensorimotor signals and produce symbolic output for robot control and human interaction. One of the main challenges for this route to succeed is to anchor its processing in verifiable symbolic structures and operations, so that any errors or hallucinations do not cause unintended harm. VLA models bypass the explicit visual understanding stage altogether and directly produce robot control commands. However, they still suffer from the same problems with LLM/VLMs, and do not solve the problem of generalization to delicate manipulation that may require tactile or force control. In all foundation model based systems, one important obstacle that must be overcome is the high energy required to train and operate such systems. In addition to the aforementioned issues, broadening the impact of neuro-symbolic robotics requires further research effort to exploit neuro-symbolic systems for robotic navigation and multi-robot systems, which are currently underrepresented.

References

[1] A. d'Avila Garcez and L. C. Lamb, "Neurosymbolic ai: The 3rd wave," *arXiv e-prints*, 2020.

[2] A. Newell, "Physical symbol systems," *Cognitive science*, vol. 4, no. 2, pp. 135–183, 1980.

[3] L. Steels, *Symbols and embodiment: Debates on meaning and cognition*. Oxford University Press Oxford, 2008, ch. The symbol grounding problem has been solved. so what's next?, pp. 223–244.

[4] T. Taniguchi, E. Ugur, M. Hoffmann, L. Jamone, T. Nagai, B. Rosman, and T. Matsuka, "Symbol emergence in cognitive developmental systems: a survey," *IEEE TCDS*, vol. 11, pp. 494–516, 2018.

[5] T. R. Besold, A. d'Avila Garcez, S. Bader, H. Bowman, P. Domingos, P. Hitzler, K.-U. Kühnberger, L. C. Lamb, P. M. V. Lima, L. de Penning *et al.*, "Neural-symbolic learning and reasoning: A survey and interpretation 1," in *Neuro-Symbolic Artificial Intelligence: The State of the Art*. IOS press, 2021, pp. 1–51.

[6] G. Marcus, "The next decade in ai: four steps towards robust artificial intelligence," *arXiv:2002.06177*, 2020.

[7] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv:2303.08774*, 2023.

[8] P. Hitzler, A. Eberhart, M. Ebrahimi, M. K. Sarker, and L. Zhou, "Neuro-symbolic approaches in artificial intelligence," *National Science Review*, vol. 9, no. 6, p. nwac035, 2022.

[9] M. K. Saker, L. Zhou, A. Eberhart, and P. Hitzler, "Neuro-symbolic artificial intelligence: Current trends," *Ai Communications*, vol. 34, no. 3, pp. 197–209, 2022.

[10] V. Cohen, J. X. Liu, R. Mooney, S. Tellex, and D. Watkins, "A survey of robotic language grounding: Tradeoffs between symbols and embeddings," *arxiv: 2405.13245*, 2024.

[11] S. Harnad, "The symbol grounding problem," *Physica D: Nonlinear Phenomena*, vol. 42, no. 1–3, p. 335–346, 1990.

[12] B. Alt, J. Dvorak, D. Katic, R. Jäkel, M. Beetz, and G. Lanza, "Bansai: Towards bridging the ai adoption gap in industrial robotics with neurosymbolic programming," *rXiv:2404.13652*, 2024.

[13] A. Gomaa, B. Mahdy, N. Kleer, M. Feld, F. Kirchner, and A. Krüger, "Teach me how to learn: A perspective review towards user-centered neuro-symbolic learning for robotic surgical systems," *arXiv:2307.03853*, 2023.

[14] N. Hemken, F. Jacob, F. Peller-Konrad, R. Kartmann, T. Asfour, and H. Hartenstein, "How to raise a robot–a case for neuro-symbolic ai in constrained task planning for humanoid assistive robots," *arXiv:2312.08820*, 2023.

[15] G. M. Tawfik, K. A. S. Dila, M. Y. F. Mohamed, D. N. H. Tam, N. D. Kien, A. M. Ahmed, and N. T. Huy, "A step by step guide for conducting a systematic review and meta-analysis with simulation data," *Tropical medicine and health*, vol. 47, p. 46, 2019.

[16] K. Krippendorff, *Content analysis: An introduction to its methodology*. Sage publications, 2018.

[17] K. Mourao, R. Petrick, and M. Steedman, "Using kernel perceptrons to learn action effects for planning," in *CogSys*, 2008, pp. 45–50.

[18] W. Yuan, C. Paxton, K. Desingh, and D. Fox, "SORNet: Spatial object-centric representations for sequential manipulation," in *CoRL*, 2022.

[19] T. Migimatsu and J. Bohg, "Grounding predicates through actions," in *ICRA*, 2022, pp. 3498–3504.

[20] M. Diehl, C. Paxton, and K. Ramirez-Amaro, "Automated generation of robotic planning domains from observations," in *IROS*, 2021, pp. 6732–6738.

[21] M. Helmert, "The fast downward planning system," *JAIR*, vol. 26, pp. 191–246, 2006.

[22] S. R. Ahmadzadeh, A. Paikan, F. Mastrogiovanni, L. Natale, P. Kormushev, and D. G. Caldwell, "Learning symbolic representations of actions from human demonstrations," in *ICRA*, 2015.

[23] S. Mukherjee, C. Paxton, A. Mousavian, A. Fishman, M. Likhachev, and D. Fox, "Reactive long horizon task execution via visual skill and precondition models," in *IROS*, 2021, pp. 5717–5724.

[24] S. Gugliermo, E. Schaffernicht, C. Koniaris, and F. Pecora, "Learning behavior trees from planning experts using decision tree and logic factorization," *IEEE RA-L*, vol. 8, no. 6, pp. 3534–3541, 2023.

[25] V. Sarathy, D. Kasenberg, S. Goel, J. Sinapov, and M. Scheutz, "SPOTTER: Extending symbolic planning operators through targeted reinforcement learning," in *AAMAS*, 2021, pp. 1118–1126.

[26] F. S. Lay, A. S. Bauer, and D. Leidner, "Unsupervised symbol emergence for supervised autonomy using multi-modal latent dirichlet allocations," *Advanced Robotics*, vol. 36, no. 1-2, pp. 71–84, 2022.

[27] T. Ahmad, L. Jin, X. Zhang, S. Lai, G. Tang, and L. Lin, "Graph convolutional neural network for human action recognition: A comprehensive survey," *IEEE TAI*, vol. 2, pp. 128–145, 2021.

[28] A. Ahmetoglu, M. Y. Seker, J. Piater, E. Oztop, and E. Ugur, "DeepSym: Deep symbol generation and rule learning for planning from unsupervised robot interaction," *JAIR*, vol. 75, pp. 709–745, 2022.

[29] G. Konidaris, L. P. Kaelbling, and T. Lozano-Perez, "From skills to symbols: Learning symbolic representations for abstract high-level planning," *JAIR*, vol. 61, pp. 215–289, 2018.

[30] T. Silver, R. Chitnis, J. Tenenbaum, L. P. Kaelbling, and T. Lozano-Pérez, "Learning symbolic operators for task and motion planning," in *IROS*, 2021.

[31] N. Shah, J. Nagpal, P. Verma, and S. Srivastava, "From reals to logic and back: Inventing symbolic vocabularies, actions and models for planning from raw data," *arXiv:2402.11871*, 2024.

[32] T. Taniguchi, T. Nagai, and Nakamura, "Symbol emergence in robotics: a survey," *Advanced Robotics*, vol. 30, pp. 706–728, 2016.

[33] L. W. Barsalou, "Perceptions of perceptual symbols," *Behavioral and brain sciences*, vol. 22, no. 04, pp. 637–660, 1999.

[34] T. Nakamura, Y. Ando, T. Nagai, and M. Kaneko, "Concept formation by robots using an infinite mixture of models," in *IROS*, 2015.

[35] T. Nakamura, T. Nagai, and T. Taniguchi, "Serket: An architecture for connecting stochastic models to realize a large-scale cognitive model," *Frontiers in Neurorobotics*, vol. 12, p. 25, 2018.

[36] T. Taniguchi, T. Nakamura, M. Suzuki, A. Taniguchi, T. Horii, and T. Nagai, "Neuro-serket: development of integrative cognitive system through the composition of deep probabilistic generative models," *New Generation Computing*, vol. 38, pp. 23–48, 2020.

[37] T. Taniguchi, H. Yamakawa, T. Nagai, K. Doya, M. Sakagami, M. Suzuki, T. Nakamura, and A. Taniguchi, "A whole brain probabilistic generative model: Toward realizing cognitive architectures for developmental robots," *Neural Networks*, vol. 150, pp. 293–312, 2022.

[38] O. B. Ozturkcu, E. Ugur, and E. Oztop, "High-level representations through unconstrained sensorimotor learning," in *ICDL*, 2020.

[39] N. Gopalan, E. Rosen, G. Konidaris, and S. Tellex, "Simultaneously learning transferable symbols and language groundings from perceptual data for instruction following," *RSS*, 2020.

[40] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *NeurIPS*, 2017.

[41] A. Ahmetoglu, B. Celik, E. Oztop, and E. Ugur, "Discovering predictive relational object symbols with symbolic attentive layers," *IEEE RA-L*, 2024.

[42] A. Ahmetoğlu, E. Öztop, and E. Uğur, "Deep multi-object symbol learning with self-attention based predictors," in *SIU*, 2023, pp. 1–4.

[43] A. Ahmetoglu, E. Oztop, and E. Ugur, "Symbolic manipulation planning with discovered object and relational predicates," *IEEE RA-L*, 2025.

[44] M. Cakmak, M. R. Dogar, E. Ugur, and E. Sahin, "Affordances as a framework for robot control," in *Epigenetic Robotics*, 2007.

[45] R. Veerapaneni, J. D. Co-Reyes, M. Chang, M. Janner, C. Finn, J. Wu, J. Tenenbaum, and S. Levine, "Entity abstraction in visual model-based reinforcement learning," in *CoRL*, 2020, p. 1439.

[46] E. Ugur, Y. Nagai, E. Sahin, and E. Oztop, "Staged development of robot skills: Behavior formation, affordance learning and imitation with motionese," *IEEE TAMD*, vol. 7, pp. 119–139, 2015.

[47] R. Chitnis, T. Silver, J. B. Tenenbaum, T. Lozano-Perez, and L. P. Kaelbling, "Learning neuro-symbolic relational transition models for bilevel planning," in *IROS*, 2022.

[48] T. Silver, A. Athalye, J. B. Tenenbaum, T. Lozano-Perez, and L. P. Kaelbling, "Learning neuro-symbolic skills for bilevel planning," in *CoRL*, 2022.

[49] J. Achterhold, M. Krimmel, and J. Stueckler, "Learning temporally extended skills in continuous domains as symbolic actions for planning," in *CoRL*. PMLR, 2023, pp. 225–236.

[50] Z. Wang, C. R. Garrett, L. P. Kaelbling, and T. Lozano-Pérez, "Active model learning and diverse action sampling for task and motion planning," in *IROS*, 2018.

[51] ——, "Learning compositional models of robot skills for task and motion planning," *IJRR*, vol. 40, 2021.

[52] C. R. Garrett, T. Lozano-Pérez, and L. P. Kaelbling, "Sample-based methods for factored task and motion planning," in *RSS*, 2017.

[53] ——, "PDDLStream: Integrating symbolic planners and blackbox samplers via optimistic adaptive planning," in *ICAPS*, 2020.

[54] H. Aktas and E. Ugur, "Vq-cnmp: Neuro-symbolic skill learning for bi-level planning," *arXiv:2410.10045*, 2024.

[55] T. Silver, R. Chitnis, N. Kumar, W. McClinton, T. Lozano-Pérez, L. Kaelbling, and J. B. Tenenbaum, "Predicate invention for bilevel planning," in *AAAI*, vol. 37, 2023.

[56] N. Kumar, W. McClinton, R. Chitnis, T. Silver, T. Lozano-Pérez, and L. P. Kaelbling, "Learning efficient abstract planning models that choose what to predict," in *CoRL*, 2023.

[57] A. Li and T. Silver, "Embodied active learning of relational state abstractions for bilevel planning," *arXiv:2303.04912*, 2023.

[58] L. Wong, J. Mao, P. Sharma, Z. S. Siegel, J. Feng, N. Korneev, J. B. Tenenbaum, and J. Andreas, "Learning adaptive planning representations with natural language guidance," *arXiv:2312.08566*, 2023.

[59] R. Rodriguez-Sanchez and G. Konidaris, "Learning abstract world models for value-preserving planning with options," *Reinforcement Learning Journal*, vol. 4, pp. 1733–1758, 2024.

[60] B. Kilic, A. Ahmetoglu, and E. Ugur, "Predictability-based curiosity-guided action symbol discovery," in *IEEE ICDL*, 2025.

[61] R. S. Sutton, D. Precup, and S. Singh, "Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning," *Artificial intelligence*, vol. 112, no. 1-2, pp. 181–211, 1999.

[62] G. Konidaris and A. Barto, "Skill discovery in continuous reinforcement learning domains using skill chaining," *NeurIPS*, vol. 22, 2009.

[63] P.-L. Bacon, J. Harb, and D. Precup, "The option-critic architecture," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, no. 1, 2017.

[64] A. Bagaria and G. Konidaris, "Option discovery using deep skill chaining," in *ICLR*, 2019.

[65] A. Bagaria, J. K. Senthil, and G. Konidaris, "Skill discovery for exploration and planning using deep skill graphs," in *ICML*, 2021, pp. 521–531.

[66] D. Molina, K. Kumar, and S. Srivastava, "Learn and link: Learning critical regions for efficient planning," in *ICRA*, 2020.

[67] N. Shah and S. Srivastava, "Using deep learning to bootstrap abstractions for hierarchical robot planning," in *AAMAS*, 2022.

[68] N. Shah, "Learning neuro-symbolic abstractions for robot planning and learning," in *AAAI*, 2024.

[69] ——, "Autonomously learning world-model representations for efficient robot planning," Arizona State University, Tech. Rep., 2024.

[70] Y. Liang, N. Kumar, H. Tang, A. Weller, J. B. Tenenbaum, T. Silver, J. F. Henriques, and K. Ellis, "Visualpredicator: Learning abstract world models with neuro-symbolic predicates for robot planning," *arXiv:2410.23156*, 2024.

[71] A. Athalye, N. Kumar, T. Silver, Y. Liang, T. Lozano-Pérez, and L. P. Kaelbling, "Predicate invention from pixels via pretrained vision-language models," *arXiv:2501.00296*, 2024.

[72] E. Saccon, A. Tikna, D. De Martini, E. Lamon, and L. Palopoli, "When prolog meets generative models: a new approach for managing knowledge and planning in robotic applications," in *ICRA*, 2024.

[73] A. Capitanelli and F. Mastrogiovanni, "A framework for neurosymbolic robot action planning using large language models," *Frontiers in Neurorobotics*, vol. 18, p. 1342786, 2024.

[74] M. Han, Y. Zhu, S.-C. Zhu, Y. N. Wu, and Y. Zhu, "Interpret: Interactive predicate learning from language feedback for generalizable task planning," *arXiv:2405.19758*, 2024.

[75] J. Sun, H. Sun, T. Han, and B. Zhou, "Neuro-symbolic program search for autonomous driving decision module design," in *CoRL*, 2021.

[76] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, and C. Finn, "Do as I can, not as I say: Grounding language in robotic affordances," in *CoRL*, 2022.

[77] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann *et al.*, "Palm: Scaling language modeling with pathways," *Journal of Machine Learning Research*, vol. 24, no. 240, pp. 1–113, 2023.

[78] G. Tziafas and H. Kasaei, "Enhancing interpretability and interactivity in robot manipulation: Neurosymbolic approach," *arXiv:2210.00858*, 2022.

[79] I. Sharifi, M. Yildirim, and S. Fallah, "Towards safe autonomous driving policies using a neuro-symbolic deep reinforcement learning approach," *arXiv:2307.01316*, 2023.

[80] C. Cornelio and M. Diab, "Recover: A neuro-symbolic framework for failure detection and recovery," *arXiv:2404.00756*, 2024.

[81] G. Chalvatzaki, A. Younes, D. Nandha, A. T. Le, L. F. Ribeiro, and I. Gurevych, "Learning to reason over scene graphs: a case study of finetuning gpt-2 into a robot language model for grounded task planning," *Frontiers in Robotics and AI*, vol. 10, p. 1221739, 2023.

[82] Y. Zhu, J. Tremblay, S. Birchfield, and Y. Zhu, "Hierarchical planning for long-horizon manipulation with geometric and symbolic scene graphs," in *ICRA*, 2021, pp. 6541–6548.

[83] A. M. Wells, N. T. Dantam, A. Shrivastava, and L. E. Kavraki, "Learning feasibility for task and motion planning in tabletop environments," *IEEE RA-L*, vol. 4, no. 2, pp. 1255–1262, 2019.

[84] D. Driess, J.-S. Ha, and M. Toussaint, "Deep visual reasoning: Learning to predict action sequences for task and motion planning from an initial scene image," in *RSS*, 2020.

[85] R. Chitnis, D. Hadfield-Menell, A. Gupta, S. Srivastava, E. Groshev, C. Lin, and P. Abbeel, "Guided search for task and motion plans using learned heuristics," in *ICRA*, 2016, pp. 447–454.

[86] T. Hiraoka, T. Onishi, T. Imagawa, and Y. Tsuruoka, "Refining manually-designed symbol grounding and high-level planning by policy gradients," *arXiv:1810.00177*, 2018.

[87] L. Lamanna, A. Saetti, L. Serafini, and A. Gerevini, "Online learning of action models for pddl planning." in *IJCAI*, 2021, pp. 4112–4118.

[88] L. Lamanna, M. Faridghasemnia, A. Gerevini, A. Saetti, A. Saffiotti, L. Serafini, P. Traverso *et al.*, "Learning to act for perceiving in partially unknown environments." in *IJCAI*, 2023, pp. 5485–5493.

[89] G. Anderson, A. Verma, and I. Dillig, "Neurosymbolic reinforcement learning with formally verified exploration," *NeurIPS*, 2020.

[90] S. Cheng and D. Xu, "League: Guided skill learning and abstraction for long-horizon manipulation," *IEEE RA-L*, 2023.

[91] T. Mota and M. Sridharan, "Commonsense reasoning and knowledge acquisition to guide deep learning on robots," in *RSS*, 2019.

[92] A. Ahmetoglu, E. Ugur, M. Asada, and E. Oztop, "High-level features for resource economy and fast learning in skill transfer," *Advanced Robotics*, vol. 36, pp. 291–303, 2022.

[93] U. Rakhman, J. Ahn, and C. Nam, "Fully automatic data collection for neuro-symbolic task planning for mobile robot navigation," in *IEEE SMC*, 2021, pp. 450–455.

[94] L. Lamanna, L. Serafini, M. Faridghasemnia, A. Saffiotti, A. Saetti, A. Gerevini, and P. Traverso, "Planning for learning object properties," in *AAAI*, vol. 37, 2023, p. 12005.

[95] P. Verma, S. R. Marpally, and S. Srivastava, "Discovering user-interpretable capabilities of black-box planning agents," in *Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning*, 2022.

[96] O. Bastani, Y. Pu, and A. Solar-Lezama, "Verifiable reinforcement learning via policy extraction," *NeurIPS*, vol. 31, 2018.

[97] G. Weiss, Y. Goldberg, and E. Yahav, "Extracting automata from recurrent neural networks using queries and counterexamples," in *ICML*, 2018, pp. 5247–5256.

[98] I. C. Kaadoud, N. P. Rougier, and F. Alexandre, "Knowledge extraction from the learning of sequences in a long short-term memory (lstm) architecture," *Knowledge-Based Systems*, vol. 235, p. 107657, 2022.

[99] S. Carr, N. Jansen, and U. Topcu, "Verifiable RNN-based policies for POMDPs under temporal logic constraints," in *IJCAI*, 2020, pp. 4121–4127.

[100] X. Sun and Y. Shoukry, "Neurosymbolic motion and task planning for linear temporal logic tasks," *IEEE Transactions on Robotics*, 2024.

[101] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu *et al.*, "RT-1: Robotics transformer for real-world control at scale," in *RSS*, 2023.

[102] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, A. Brohan, N. Brown *et al.*, "RT-2: Vision-language-action models transfer web knowledge to robotic control," in *CoRL*, 2023, pp. 2165–2183.

[103] A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee *et al.*, "Open X-embodiment: Robotic learning datasets and RT-X models," in *ICRA*, 2024, pp. 6892–6903.

[104] H. Arai, K. Miwa, K. Sasaki, Y. Yamaguchi, K. Watanabe, S. Aoki, and I. Yamamoto, "Covla: Comprehensive vision-language-action dataset for autonomous driving," *arXiv:2408.10845*, 2024.

[105] K. Kawaharazuka, J. Oh, J. Yamada, I. Posner, and Y. Zhu, "Vision-language-action models for robotics: A review towards real-world applications," *IEEE Access*, 2025.

[106] R. Wang, J. Mao, J. Hsu, H. Zhao, J. Wu, and Y. Gao, "Programmatically grounded, compositionally generalizable robotic manipulation," *arXiv:2304.13826*, 2023.

[107] V. Pallagani, B. Muppasani, B. Srivastava, F. Rossi, L. Horesh, K. Murugesan, A. Loreggia, F. Fabiano, R. Joseph, Y. Kethepalli *et al.*, "Plansformer tool: Demonstrating generation of symbolic plans using transformers." in *IJCAI*, 2023, pp. 7158–7162.

[108] K. Namasivayam, H. Singh, V. Bindal, A. Tuli, V. Agrawal, R. Jain, P. Singla, and R. Paul, "Learning neuro-symbolic programs for language guided robot manipulation," in *ICRA*, 2023.

[109] J. Mao, C. Gan, P. Kohli, J. B. Tenenbaum, and J. Wu, "The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision," in *ICLR*, 2019.

[110] Y. Lin, A. S. Wang, E. Undersander, and A. Rai, "Efficient and interpretable robot manipulation with graph neural networks," *IEEE RA-L*, vol. 7, no. 2, pp. 2740–2747, 2022.

[111] M. Zadem, S. Mover, and S. M. Nguyen, "Goal space abstraction in hierarchical reinforcement learning via reachability analysis," in *ICDL*, 2023.

[112] ——, "Reconciling spatial and temporal abstractions for goal representation," *ICLR*, 2024.
[113] M. Zadem, S. Mover *et al.*, "Emergence of a symbolic goal representation with an intelligent tutoring system based on intrinsic motivation," in *NeurIPS2023 Workshops*, 2023, pp. 423–428.
[114] E. Heiden, D. Millard, E. Coumans, Y. Sheng, and G. S. Sukhatme, "Neuralsim: Augmenting differentiable simulators with neural networks," in *ICRA*, 2021, pp. 9474–9481.
[115] S. Sanyal and K. Roy, "Ramp-net: A robust adaptive mpc for quadrotors via physics-informed neural network," in *ICRA*, 2023, pp. 1019–1025.
[116] J. H. Li and P. M. Lee, "A neural network adaptive controller design for free-pitch-angle diving behavior of an autonomous underwater vehicle," *Robotics and Autonomous Systems*, vol. 52, no. 2-3, pp. 132–147, 2005.
[117] J. Pisokas and U. Nehmzow, "Experiments in subsymbolic action planning with mobile robots," in *AAMAS*, 2003.
[118] E. Ugur, E. Sahin, and E. Öztop, "Affordance learning from range data for multi-step planning." in *EpiRob*, 2009.
[119] E. Ugur, Şahin, and E. Oztop, "Unsupervised learning of object affordances for planning in a mobile manipulation platform," in *ICRA*, 2011.
[120] J. Mugan and B. Kuipers, "Autonomous learning of high-level states and actions in continuous environments," *IEEE Transactions on Autonomous Mental Development*, vol. 4, no. 1, pp. 70–86, 2011.
[121] T. Nakamura, T. Nagai, and N. Iwahashi, "Multimodal categorization by hierarchical dirichlet process," in *IROS*, 2011, pp. 1520–1525.
[122] Y. Ando, T. Nakamura, T. Araki, and T. Nagai, "Formation of hierarchical object concept using hierarchical latent Dirichlet allocation," in *IROS*, 2013, pp. 2272–2279.
[123] T. Nakamura, T. Nagai, and N. Iwahashi, "Bag of multimodal LDA models for concept formation," in *ICRA*, 2011, pp. 6233–6238.
[124] ——, "Bag of multimodal hierarchical dirichlet processes: Model of complex conceptual structure for intelligent robots," in *IROS*, 2012, pp. 3818–3823.
[125] T. Araki, T. Nakamura, T. Nagai, S. Nagasaka, T. Taniguchi, and N. Iwahashi, "Online learning of concepts and words using multimodal LDA and hierarchical Pitman-Yor language model," in *IROS*, 2012, pp. 1623–1630.
[126] T. Nakamura, T. Nagai, K. Funakoshi, S. Nagasaka, T. Taniguchi, and N. Iwahashi, "Mutual Learning of an Object Concept and Language Model Based on MLDA and NPYLM," in *IROS*, 2014, pp. 600 – 607.
[127] J. Nishihara, T. Nakamura, and T. Nagai, "Online algorithm for robots to learn object concepts and language model," *IEEE TCDS*, vol. 9, no. 3, pp. 255–268, 2016.
[128] R. Kuniyasu, T. Nakamura, T. Taniguchi, and T. Nagai, "Robot concept acquisition based on interaction between probabilistic and deep generative models," *Frontiers in Computer Science*, vol. 3, p. 618069, 2021.
[129] A. Taniguchi, T. Taniguchi, and T. Inamura, "Spatial concept acquisition for a mobile robot that integrates self-localization and unsupervised word discovery from spoken sentences," *IEEE TCDS*, vol. 8, no. 4, pp. 285–297, 2016.
[130] A. Taniguchi, Y. Hagiwara, T. Taniguchi, and T. Inamura, "Online spatial concept and lexical acquisition with simultaneous localization and mapping," in *IROS*, 2017, pp. 811–818.
[131] A. Taniguchi, T. Taniguchi, and T. Inamura, "Unsupervised spatial lexical acquisition by updating a language model with place clues," *Robotics and Autonomous Systems*, vol. 99, pp. 166–180, 2018.
[132] R. Sagara, R. Taguchi, A. Taniguchi, T. Taniguchi, K. Hattori, M. Hoguro, and T. Umezaki, "Unsupervised lexical acquisition of relative spatial concepts using spoken user utterances," *Advanced Robotics*, vol. 36, no. 1-2, pp. 54–70, 2022.
[133] S. Hasegawa, A. Taniguchi, Y. Hagiwara, L. El Hafi, and T. Taniguchi, "Inferring place-object relationships by integrating probabilistic logic and multimodal spatial concepts," in *SII*, 2023, pp. 1–8.
[134] E. Rosen, S. James, S. Orozco, V. Gupta, M. Merlin, S. Tellex, and G. Konidaris, "Synthesizing navigation abstractions for planning with portable manipulation skills," in *CoRL*, 2023, pp. 2278–2287.
[135] A. Pacheck, S. James, G. Konidaris, and H. Kress-Gazit, "Automatic encoding and repair of reactive high-level tasks with learned abstract representations," *IJRR*, vol. 42, 2023.
[136] E. Ugur and J. Piater, "Bottom-up learning of object categories, action effects and logical rules: From continuous manipulative exploration to symbolic planning," in *ICRA*, 2015, pp. 2627–2633.
[137] ——, "Refining discovered symbols with multi-step interaction experience," in *umanoids*, 2015, pp. 1007–1012.
[138] N. Jetchev, T. Lang, and M. Toussaint, "Learning grounded relational symbols from continuous data for abstract reasoning," in *Proceedings of the 2013 ICRA Workshop on Autonomous Learning*, 2013.
[139] B. Ames, A. Thackston, and G. Konidaris, "Learning symbolic representations for planning with parameterized skills," in *IROS*, 2018, pp. 526–533.
[140] G. Konidaris, L. Kaelbling, and T. Lozano-Perez, "Constructing symbolic representations for high-level planning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 28, no. 1, 2014.
[141] G. Konidaris, L. P. Kaelbling, and T. Lozano-Perez, "Symbol acquisition for probabilistic high-level planning," in *AAAI*, 2015.
[142] S. James, B. Rosman, and G. Konidaris, "Learning portable representations for high-level planning," in *ICML*, 2020, pp. 4682–4691.
[143] ——, "Autonomous learning of object-centric abstractions for high-level planning," in *ICLR*, 2022.
[144] D. Jung and A. Zelinsky, "Grounded symbolic communication between heterogeneous cooperating robots," *Autonomous Robots*, vol. 8, no. 3, pp. 269–292, 2000.
[145] C. Paxton, A. Hundt, F. Jonathan, K. Guerin, and G. D. Hager, "Costar: Instructing collaborative robots with behavior trees and vision," in *ICRA*, 2017, pp. 564–571.
[146] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," *NeurIPS*, 2022.
[147] S. Yao, D. Yu, J. Zhao, I. Shafran, T. Griffiths, Y. Cao, and K. Narasimhan, "Tree of thoughts: Deliberate problem solving with large language models," *NeurIPS*, vol. 36, 2024.
[148] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "React: Synergizing reasoning and acting in language models," in *ICLR*, 2023.
[149] T. Schick, J. Dwivedi-Yu, R. Dessì, R. Raileanu, M. Lomeli, E. Hambro, L. Zettlemoyer, N. Cancedda, and T. Scialom, "Toolformer: Language models can teach themselves to use tools," *NeurIPS*, vol. 36, pp. 68 539–68 551, 2023.
[150] D. TUCKER, "Intensionality and paradoxes in ramsey's 'the foundations of mathematics'," *The Review of Symbolic Logic*, vol. 3, no. 1, p. 1–25, 2010.
[151] A. Lazaridou and M. Baroni, "Emergent multi-agent communication in the deep learning era," *arXiv:2006.02419*, 2020.
[152] L. Steels, "Modeling the cultural evolution of language," *Physics of Life Reviews*, vol. 8, 2011.
[153] C. Zhu, M. Dastani, and S. Wang, "A survey of multi-agent deep reinforcement learning with communication," *AAMAS*, vol. 38, no. 1, p. 4, 2024.
[154] J. Peters, C. W. de Puiseau, H. Tercan, A. Gopikrishnan, G. A. L. D. Carvalho, C. Bitter, and T. Meisen, "A survey on emergent language," *arxiv:2409.02645*, 2024.
[155] R. Manhaeve, S. Dumancic, A. Kimmig, T. Demeester, and L. De Raedt, "Deepproblog: Neural probabilistic logic programming," *NeurIPS*, vol. 31, 2018.
[156] S. Badreddine, A. d. Garcez, L. Serafini, and M. Spranger, "Logic tensor networks," *Artificial Intelligence*, vol. 303, p. 103649, 2022.
[157] Z. Yang, A. Ishay, and J. Lee, "Neurasp: Embracing neural networks into answer set programming," *arXiv:2307.07700*, 2023.
[158] P. Lorang, H. Lu, J. Huemer, P. Zips, and M. Scheutz, "Few-shot neuro-symbolic imitation learning for long-horizon planning and acting," *arXiv:2508.21501*, 2025.
[159] T. Taniguchi, Y. Hirai, M. Suzuki, S. Murata, T. Horii, and K. Tanaka, "System 0/1/2/3: Quad-process theory for multi-timescale embodied collective cognitive systems," *arXiv:2503.06138*, 2025.
[160] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, "Code as policies: Language model programs for embodied control," *arXiv:2209.07753*, 2022.
[161] A. d. Garcez, T. R. Besold, L. De Raedt, P. Földiak, P. Hitzler, T. Icard, K.-U. Kühnberger, L. C. Lamb, R. Miikkulainen, and D. L. Silver, "Neural-symbolic learning and reasoning: contributions and challenges," in *AAAI*, 2015.
[162] H. Kautz, "The third ai summer: Aaai robert s. engelmore memorial lecture," *Ai magazine*, vol. 43, no. 1, pp. 105–125, 2022.
[163] P. Hitzler and M. K. Sarker, *Neuro-symbolic artificial intelligence: The state of the art*, 2022.