Neuro-Symbolic Robotics

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

***Abstract*—This article introduces and summarizes the emerging field of Neuro-Symbolic Robotics. 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 demands properties like verifiability, explainability, and interpretability, which Neural Networks lack. Furthermore, neural network-based models experience difficulties with generalization and extrapolation, thus restricting their use. Historically, symbolic systems have been integral to intelligent robotics due to their verifiability, explainability, and scalability, though their manually programmed frameworks fail to manage the complexity and diversity of the robot's continuous and high-dimensional environments effectively. This paper examines various robotic architectures that combine neural networks with symbolic systems in diverse manners to leverage their distinct advantages. We classify these robotic systems into four main categories: intertwined, coupled, non-uniform neuro-robotic systems, and neuro-symbolic translation. We provide an in-depth analysis of the strengths and weaknesses of these systems and outline the future challenges in this domain.**

***Index Terms*—Neuro-Symbolic AI, Neuro-symbolic Artificial Intelligence**

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 but yet adaptable, thereby enabling high performance in complex tasks as well as allowing reasoning for safety and explainability.

In contrast to domain-specific machine learning problems such as classification and regression, a robot needs to process a continuous stream of sensorimotor data yet act on a world that is structured as a network of discrete entities with a range of relations among them. Thus, intelligent robots need cognitive mechanisms to work with 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].

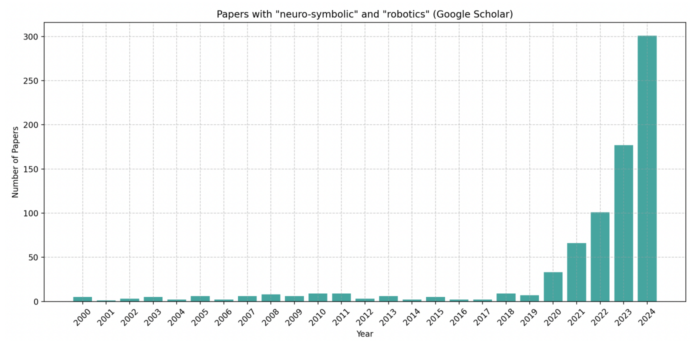

Fig. 1. Neuro-Symbolic Robotics related publication histogram, obtained via Google Scholar scraping. The data shows significant growth from 2021.

Learning Systems. The groundbreaking advances in deep neural networks and their effective application in artificial intelligence raise the question of whether deep learning with 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 allow 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 reward labeling for valid chain-of-thought generation. Thus, in

This work was supported by the INVERSE project (101136067) funded by the European Union.

[1]Bogazici University [2]Brown University [3]The University of Tokyo [4]Kyoto University [5]University of Trento [6]Osaka/Ozyegin University

[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]. In this paper, 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 the paper authored by some of the present authors [4].

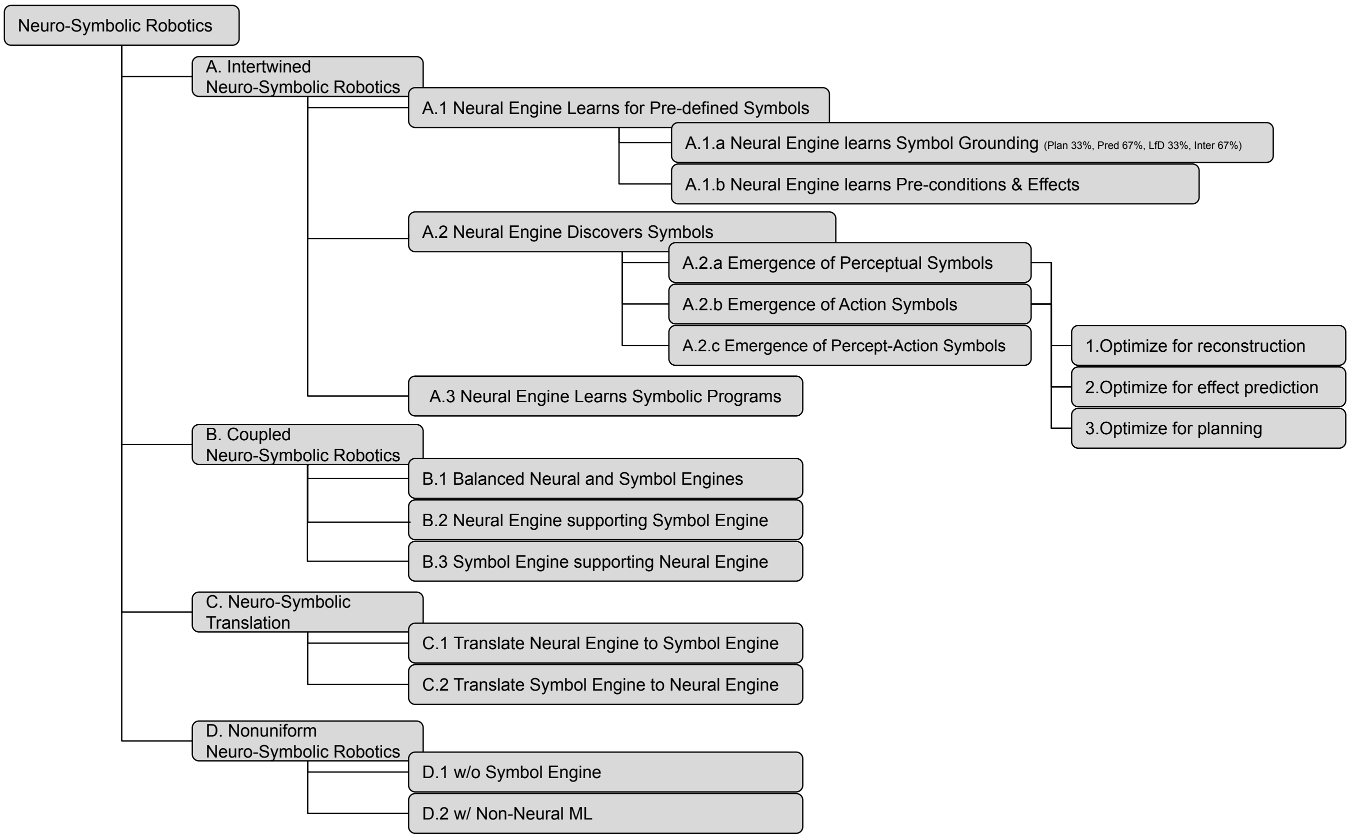

Fig. 2. A taxonomy of Neuro-Symbolic Robotics

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 full functionality and representational power of first-order logic, according to Marcus [6]. Thus, albeit the impressive reasoning-like abilities of LLMs, it is not clear how formally defined computational semantics can be neurally embedded in the operation of LLMs to address the 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 classical example of the problem of pre-defining a set of symbols and rules to represent the sensorimotor experience of a robot, which is often called 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 allow reasoning only in 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 that are used in robotics literature are delineated in [10] within the context of natural language representation, a theory of multi-level symbolic manipulation bridging the 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 not only used in symbolic manipulation systems for reasoning and planning but also equip the robot with different capabilities or improve the existing ones.

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 full-power of both neural and symbolic systems are still missing. In the remaining of the paper, we systematically analyze key robotic works from the literature that have used symbolic and learning systems together with a varying extent.

The analysis presented in this article is guided by the taxonomy given in Fig. 2 which categorizes robotic studies into four main groups based on the interplay between the neural and symbolic engines used. In the first category, the neural and symbolic engines are intertwined in such a way that the system can only operate with both engines running. Most methods in this category utilize the neural engine to generate representations that are to be 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 main characteristic of this group is that the main engine can work in isolation, and running both engines usually increases the performance. Methods in the third category are the translation-based systems that either translate the symbolic rules into neural-processable data, or distill the information in a neural engine into symbolic knowledge, allowing interpretability. To be precise, the methods falling into the fourth category do not possess a neural or symbolic framework that would qualify them as genuine neuro-symbolic methods. Nevertheless, they are part of this review because they are precursors of their neural counterparts and due to their potential for adaptation into such methods.

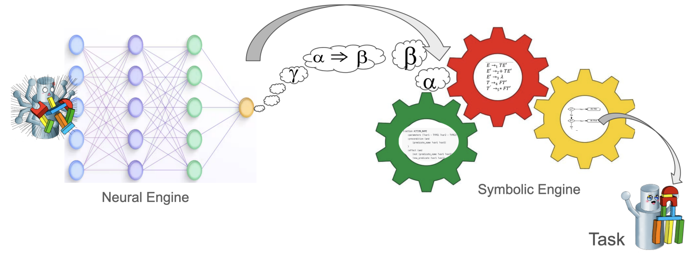

Fig. 3. The relation between Neural and Symbolic Engines in Intertwined Neuro-Symbolic Robotics. In these systems, the robot's continuous interaction experience is used by the Neural Engine to learn discrete symbols and/or rules and propagated to the Symbolic Engine for high-level reasoning.

II. Definitions

Symbolic Representation corresponds to the encoding of robot perception, action, or state in discrete space. **Continuous Representation** corresponds to continuous encoding of robot perception, action, or state that might be used as input and/or output of a Neural Network system.

Symbol Engine corresponds to the methods and algorithms used for manipulating symbols. It might correspond to classifiers such as decision trees, Monte-Carlo search trees used for multi-step prediction, operations over Domain Specific Language (DSL), or full-fledged off-the-shelf AI planning in standard symbolic languages such as Planning Domain Description Language (PDDL) [15].

Neural Engine corresponds to Neural Network used in discriminatory or generation tasks. The inputs and outputs of the Neural Engine, as well as the intermediate representations, might be discrete or continuous valued depending on the task.

III. A. Intertwined Neuro-Symbolic Robotics

In this category, the representations, rules, or programs used by the Symbolic Engine are generated by the Neural Engine. A key distinction among these types of approaches is whether program generation is at the core, or the symbol discovery is undertaken by the neural system or not. Accordingly, we have three main subcategories, which are detailed next.

A.1 Neural Engine Learns for Pre-defined Symbols

In this category, the symbols and the operators used by the Symbol Engine are pre-defined based on the task and domain requirements. The Neural Engine either learns the mapping between these discrete symbols and the continuous sensorimotor experience of the robot or the set of symbolic pre-conditions and effects of the operators 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 [16]. More recently, in [17], the system 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. Given a robot interaction video dataset with annotated actions and manually implemented pre-conditions and effects, [18] trained classifiers that map the bounding boxes of objects to the corresponding symbols. When applicable, the approach taken in this category is effective, but the fixed set of predicates and pre-defined rigid rules and operators limit 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 pre-defined transition rules used can be relaxed by learning the set of predicates for the pre-conditions and post-conditions 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 before, and the robot basically learns the set of predicates for pre-conditions and effects of actions. With this approach, [19] learned a list of pre-condition and effect predicates, first segmenting the human demonstrations into actions, then extracting the relevant pre-conditions and post-conditions based on counting heuristic, and finally generating the related planning operators. In the end, an externally given goal can be satisfied by a sequence of actions using Fast Downward PDDL planner [20]. [21] 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. Given pre-defined symbolic predicates, [22] learns a set of parameterized actions, with their corresponding pre-condition and effect predicates in a manipulation domain. [23], on the other hand, proposed to learn the planning domaies from the observed traces using Behavior Trees as intermediate human-readable structures. Given a symbolic goal, [24] learned the necessary symbolic operators to be able to synthesize a plan and its low-level controller implementation in an RL framework. Following a different approach, [25] used preconditions and effect symbols to detect task-specific deficiencies and support humans in action feasibility, rather than plan generation. These studies are constrained to a predefined set of symbols, whereas the robots might need to invent new symbols in changing environment conditions and new environments. In the next section, we cover the neuro-symbolic studies that address unsupervised or self-supervised discovery of such symbols.

A.2. Neural Engine Discovers Symbols

In this section, we cover the studies where Neural Engine is used to discover perceptual (e.g., [26]), action (e.g., [27], [28]), or sensorimotor (e.g., [29]) 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 [30] 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 the continuous perceptual space to find discrete symbolic categories using different approaches and metrics, as follows.

a) A.2.a.1 Optimize PSE for reconstruction: From the perspective of perceptual symbol systems [31], 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. For minimizing reconstruction error, probabilistic generative models (PGMs), which incorporate discrete categories as latent variables to predict multi-modal observations, have been actively pursued from the early to mid-2010s [32]. These studies, in general, clustered sensorimotor data, formed categories, and connected these categories to symbolic representations. symbol emergence in robotics has evolved into increasingly complex architectures 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 to facilitate modular and scalable implementation of these systems [33], [34]. This approach has led to the concept of a large-scale cognitive architecture, namely a whole-brain probabilistic generative model [35].

The studies above focused on probabilistic learning and inference; however, they did not fully exploit Neural Engines within their frameworks. On the other hand, [36] 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. They showed that even without explicit design or engineering, neural responses that resemble abstract symbol-like representations might emerge in their system. Recently [37] proposed to first discover skill segments from demonstration trajectories and then apply unsupervised clustering and SVM classification to identify and learn the mapping of the potential termination states of each learned skill. Eventually, these are used to learn the relation between natural language sentences and sequences of the learned abstract symbols using 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 next subsections, we cover the methods that scaffold the categorization process by taking into account the effectiveness of the symbols with respect to the task goals.

A.2.a.2 Optimize PSE for action effect prediction: The previous unsupervised clustering approach finds discrete symbols without any guarantee of being useful for the Symbol Engine. To address this problem, a number of research groups investigated how to discover symbols that are guaranteed to be useful for one-step action effect prediction, which is the most basic step of symbolic planning. [26] proposed and realized a general neural framework, namely DeepSym, that translates the robot's raw sensorimotor experience into the symbolic domain. 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 table-top environment, where symbols such as pushable, rollable, or insertable were discovered and used to make effective plans, for example, to build towers of varying heights by manipulating the objects in the environment. A limitation of the work [26] 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. Towards addressing this issue, DeepSym was extended by incorporating a transformer [38] based structure that learns what object features had to be attended through a set of attention weights [39], [40]. As in DeepSym [26], these weights were channeled through a discrete activation layer that generates relational symbols that capture 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, [41] enabled use of off-the-shelf planners by converting the discovered symbolic interactions into PDDL operators. This study showed that complex plans involving multiple objects can be made generated on the symbols discovered by the system, thereby addressing object affordances [42] to some extent, however the generalization based on affordances of complex shaped objects have 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 important to study how to maximize the utility of the discovered symbols in improving long-horizon planning performance. In this vein, [43] proposed to learn abstract relational symbolic object representations from visual observations in an unsupervised way and use them to make multi-step plans. 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 make symbolic plans with the discovered object symbols effectively, 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, robot's interactions and its physical environment.

A.2.b Action Symbol Emergence (ASE): The studies introduced up to now 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 [44] through interactive learning. In this section, we review the studies where action symbols were discovered and used in Symbol Engines.

A.2.b.1 Optimize ASE for action effect prediction: [28] proposed to formalize operator learning problem in the Task and Motion Planning (TAMP) framework. The proposed system learns operators on previously acquired action symbols, which can be defined as a lossy abstraction of the transition model of the domain. Following [28], [45] 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. 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 were 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 [46], these skills were learned from low-level demonstrations as well, providing a complete bilevel operator learning stack.

In [47] a hierarchical RL framework is adopted and a diverse set of actions are discovered while simultaneously learning symbolic forward models through intrinsic motivation signals given pre-defined state abstractions. As the Symbol Engine, the system used the breadth-first search method where each expansion corresponded to the learned symbolic forward model and executed one-by-one in order to reach the goal. In [48], [49], it is proposed to learn constraints that address the effectiveness of actions using Gaussian Processes. They proposed a sampling method for creating a rich set of potential action parameters along with the 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 [50], [51]. Last but not least, [52] discovers action symbols from human demonstrations and exploits visual language models (VLMs) not only to label those actions but also to generate plans through their scene interpretation and reasoning capabilities.

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, [53] learned symbolic predicates with a surrogate objective for multi-step planning. They used interactions obtained from demonstrations rather than the robot's own self-exploration experience of the world. In follow-up work, [54] reduced the complexity of the learned operators by focusing on a subset of abstract effects. These studies not only learn action symbols but also find the motion parameters that would allow TAMP. Although symbolic predicates are learned as well, these are defined over already available high-level predicates, which might not be realistic to assume in life-long scenarios. Following [53], [55] proposed to learn predicates by actively collecting information by querying an expert. [56] jointly learned a set of symbolic action abstractions and their low-level controllers utilizing LLMs in an interactive planning loop.

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 [57]–[61] that focus on learning these low-level policies directly from raw sensory space by exploration. While these methods do not directly use a symbol engine, they provide a finite number of low-level policies with their initiation and termination conditions overlapped in the state space, which makes the ground for action and state symbol learning.

A.2.c Percept-Action Symbol Emergence (PASE): While the previous studies focused on discovering either perceptual or action symbols, some recent studies investigated the possibility of creating perceptual and action symbols together from the sensorimotor experience of the robot [29]. [62], [63] used critical regions [64], [65], high-density regions over the state space, as state abstraction targets. 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, [66], [67] extended [53] by first learning high-level predicates directly from raw state representations using visual language models (VLMs) and then learning operators defined over these predicates.

A.3 Neural Engine learns Symbolic Programs

Within this category, all symbolic programs are crafted by the Neural Engine—predominantly using Large Language Models (LLMs)—and subsequently processed by the Symbol Engine. In the work of [68], a Neural Engine (LLM) was

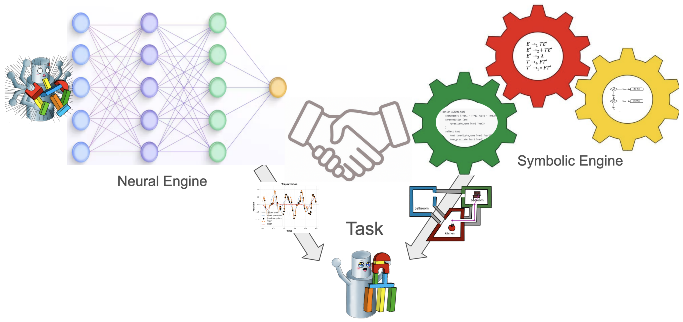

Fig. 4. The relation between Neural and Symbolic Engines in Coupled Neuro-Symbolic Robotics. In a nutshell, they can run independently and support each other for robot control.

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. [69] trained LLMs to produce neuro-symbolic task planners, which are consistent with PDDL syntax. With this, they obtained better scalability when the domain complexity was increased. This approach also enabled the production and execution of actions without waiting for the whole plan to be generated. In [70], 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, [71] 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. With the recent advances in LLMs, we expect to see more studies in this category. 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].

Discussion

The intertwined neuro-robotics studies tightly integrate Neural and Symbol Engines, such that the Symbol Engines are designed to operate only with the representations or rules that are required 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 are used less frequently. The studies that learn pre-defined symbols (A.1) or discover symbols (A.2) do not generally go beyond planning and are not used for natural language communication. The studies that learn symbolic programs (A.3), on the other hand, integrate recent LLM modules and, therefore, can address natural language. However, few of them discuss verifiability issues. Last but not least, they mostly use a given set of symbols, which limits their general use.

IV. B. Coupled Neuro-Symbolic Robotics

In this section, we overview the robotic systems where Neural Engine and Symbol Engine modules interact with each other by combining their outputs or by supporting each other. 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 [72] has been one of the first studies that benefited from the reasoning capabilities of LLMs for robot control. Given a goal, Say-Can used a language model (PALM [73]), which provided high-level semantic knowledge about a given task and generated a list of actions to achieve the task. 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 [74], 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 such primitives. Note that some primitives were symbolic (e.g., counting), and others were implemented with neural networks (e.g., visual grounding). [75] 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, [76] integrated logical rules, ontologies, and LLM-based planners and exploited symbolic information to improve the ability of LLMs to generate recovery plans. Given instructions, their robot started executing the actions for the plan generated by 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 case of failure, the ontology was again used to determine the recovery strategy, which was provided to the LLM planner for re-planning. In [77], 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. [78] combined symbolic and geometric scene graphs for vision-based long-horizon hierarchical planning. A symbolic scene graph was used to find the next sub-goal from the goal description, and the geometric scene graph was used to predict the motion parameters based on the objects embedded in the scene graph.

B.2 Neural Engines supporting Symbol Engines

In this category, the main control is on the Symbol Engine, and the Neural Engine is used to support the Symbol Engine in executing the tasks. [79] proposed to train neural network classifiers to forecast the viable motions and employ the classifier as a learned heuristic, steering the TAMP search toward possible motions and decreasing the total amount of motion planning trials. [80] proposed a Neural Engine that uses an initial image of the environment, predicting the promising sequence of discrete actions providing runtime improvements of several magnitudes. Given expert demonstrations, [81] applied learning techniques to efficiently search in the high-level task planning space, taking into account the possible infeasibilities and, as a result, significantly increasing the planning speed of the Symbol Engine. [82] 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. [83] learned relational state representations, transition function, grounding function, and action-value functions to support the planning experience of a robot. PDDL planning was used in the exploration of the agent. Learning allowed the agent to scale up to larger environments. [84] used the predicted confidence values from a Neural Engine to infer probabilistic belief states that were used by the Symbol Engine.

B.3 Symbol Engines supporting Neural Engines

In this category, the Symbol Engine is used to support the Neural Engine, which acts as the main controller. [85] used a cascade of systems in which the domain knowledge encoded by a decision tree is compiled into an Answer Set Prolog (ASP) program that classifies images and, failing to do so, redirects the input to a convolutional network. [86] used the principles of maximum information compression and slowly varying signals to extract symbol-like representations that enable fast skill transfer. 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. [87] 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. [88] also proposed a method that used Symbol Engine to decide exploratory actions for training the Neural Engine in a simulated mobile robot. [89] proposed using PDDL Symbol Engine to improve the neural perceptual capabilities of the agent by selecting which objects to observe, which properties to use, and when to stop data gathering, improving its ability to recognize new object properties.

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.

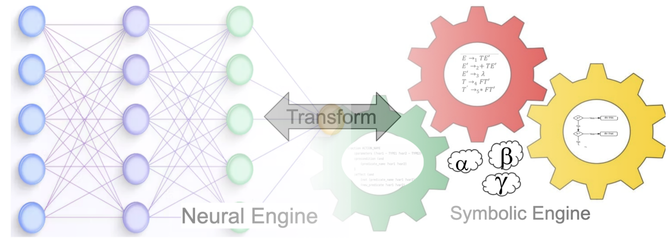

Fig. 5. The illustration of engine-to-engine transformation in the Neuro-Symbolic Translation studies.

In many of these studies, multi-step planning is achieved through PDDL-like languages. Most of these studies use simulations for learning. They verify their systems mostly in 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 support 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. 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.

V. 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. For example, [90] 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. For explainability and verifiability, [91] trained verifiable policies encoded with decision trees and realized their framework in the cart-pole task in an RL setting. Given a trained RNN, [92] learned an abstraction and extracted a deterministic finite automaton that encodes the state dynamics of the task. For transparency, trust, explainability, and interoperability, [93] applied clustering in the latent space of the Long-Short Term Memory (LSTM) network, activated by each input sequence. 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, e.g., safety and/or reachability, [94] proposed a method to autonomously build a finite-state machine from a recurrent neural network, accommodating existing formal verification tools in agent benchmarks.

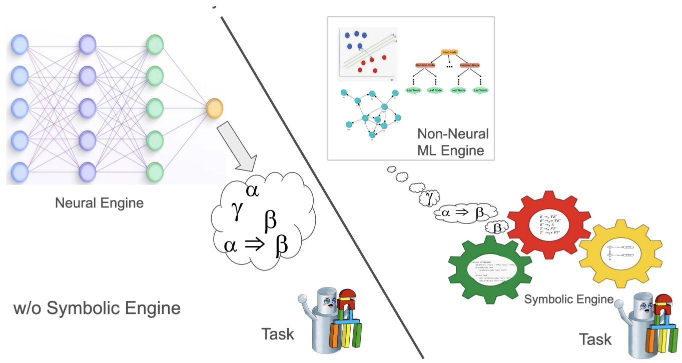

Fig. 6. Nonuniform Neuro-Symbolic Robotics.

C.2 Transform Symbol Engine to Neural Engine

In this category, the manually encoded symbolic policies are transformed into continuous neural policies and refined through the robot's experience. [95] incorporated a Symbol Engine, which used 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. The resulting Neural Network-based planner inherited the symbolic model's interpretability and correctness assurances, with the aim of generalization to unseen tasks, including new workspaces, novel temporal logic formulas, and errors in the robot's dynamical model. [96] used a Symbol Engine to realize symbolic policies and another Symbol Engine for formal verification for 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 that were updated with reward-based gradient learning and transformed back to the original symbol space. We expect an increase in the adoption of the methods that transform one engine to the other one in more robotic tasks to ensure verifiability and interoperability.

Discussion

The studies we reviewed in this section have the common aim of generating explainable systems. These approaches do not make explicit use of PDDL-like languages. As such, they do not benefit from LLMs (maybe partly because they predate the use of LLMs in robotics). The systems that translate symbolic policies into neural policies (C.2) lose multi-step planning capabilities and are rarely realized 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.

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 in these studies.

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.

[97] extended [98] to the robotics domain and proposed a "Neuro-Symbolic program" which processes both continuous and symbolic representations using trained Neural Engines. It trains a Neural Engine to parse natural language instructions, transforming it into a program in Domain Specific Language (DSL), and 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. No Symbol Engine is used in this approach.

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 [99], RT-2 [100], and RT-X [101]. The foundation models for robots, which integrate vision, language, and action, are referred to as vision-language-action (VLA) models [102]–[105]. More informally, such foundation models are also referred to as robot foundation models [106].

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. [107] used pre-trained vision language models by exploiting the semantic and syntactic relationships, disentangling action and perception, and producing control parameters for given manipulation primitives. [108] 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.

These studies are included in this survey as the Neural Engine takes goals represented as symbols and can generate intermediate steps in symbolic form. On the other hand, these studies do not benefit from Symbol Engines, therefore the generated plans are neither explainable nor verifiable directly.

[109] learned graph neural network (GNN), whose nodes encode task and domain-related entities, such as objects and outcomes, to discover rules from demonstrations. 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". [110] proposed a developmental progression for symbolic sub-goal discovery in a hierarchical

RL framework that combines together the states that have similarities for the given tasks. [111], [112] extended this work to learn both spatial and temporal goal symbols. 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.

D.2 With Non-Neural ML

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

Initial studies addressed learning sub-symbolic structures that were useful in planning. In the seminal work of [113], 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, [114], [115] first applied clustering in the effect space in manipulation and mobile manipulation domains. Their system operated by first finding effect categories and then learning SVM classifiers that map environment features to effect categories, effectively forming action-effect predictors that were used for planning via tree-search algorithms. [116] learned discrete representations from environment features and a set of predictive models based on these discrete environment symbols, enabling a simulated manipulation robot to move blocks in specified directions or pick them up and hit them to the floor. The predictive model is represented by Dynamic Bayesian Networks (DBNs), which are converted into symbolic plans to generate and execute a sequence of actions. 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 methods 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 [117], hierarchical [118], and modality-weighted [119], [120] multi-modal object categories [32]. Furthermore, [121] introduced an unsupervised word segmentation method capable of discovering words in an unsupervised manner alongside MLDA-based multi-modal object categorization. Nakamura et al. [122] and Nishihara et al. [123] 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, [124] explored object category discovery and multi-modal symbol learning using Modal Latent Dirichlet Allocation (MLDA) and variational autoencoders. 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 [125]–[128]. 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. [129] presented an early example of a neuro-symbolic approach integrating spatial concepts with probabilistic logic.

Konidaris et al. [27], [130], [131] discussed that high-level planning can be achieved by learning discrete symbols that were used to encode the preconditions and the post-conditions of the action repertoire of agents. Again, as a precursor of neural-network-based approaches, they learned symbols to encode action preconditions and postconditions and used them in the operators for building a PDDL description of the environment of the learning agent. Ames et al. learned symbolic operators along with skill parametrization in [132]. By pre-processing an incoming image with independent component analysis followed by image-to-symbol mapping via Support Vector Machines, [133] proposed a method to form symbols from raw images, following [27]. Then, the given robot skills were encoded in Linear Temporal Logic (LTL), enabling symbolic planning for tasks written using LTL formulas.

In these works, the state of each object was represented with a fixed-sized vector, assuming that the number of objects was the same across different environments and tasks. [134] 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 the follow-up work, [135] showed that generalization capability can be increased through the use of object-centric representations. [136], [137] also learned object-centric symbols used in preconditions and effects of PDDL operators. Commonly, to find discrete representation, the first step in these studies was to apply clustering on the observed interaction instances. 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.

[138] proposed an RL-based symbol learning framework where learned symbolic relational abstractions are used for encoding transition and reward model, action effect prediction, and finally, multi-step planning. The optimization of symbol learning focuses not only on enhancing effect prediction performance but also on maximizing rewards. The Nearest Neighbor method was used to learn feature-symbol mapping in this work. [139] 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 allows planning using only manipulation skills and then filling out the navigation steps automatically.

Discussion

In this section, we reviewed a mixed body of literature covering architectures that did not explicitly use either symbolic or neural engines but 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 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, use LLMs to do multi-step planning, mostly learn from human demonstrations, do not use active learning or intrinsic motivation, and do not often discuss explainability or verifiability.

VI. General Discussion

A. Neuro-Symbolic Robotics in the Age of LLMs

Recent advances in large language models (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) [140], Tree-of-Thought (ToT) [141], Self-Refine [142], and Reflexion [143]. Additionally, neuro-symbolic approaches such as ReAct [144] and ToolFormer [145] 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 showed us that non-trivial computation can be undertaken even without such guarantees if we were 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 which 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 [146]. 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 one hand, LLMs use formal symbols as external tools to enhance 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 [147]–[150]. Future research should explore ways to align the workings of LLMs with an underlying internal formal system to pave the way toward trustworthy and explainable AI systems that can scale.

B. Capabilities and Challenges

In this paper, we introduce the field of Neuro-Symbolic Robotics and create a taxonomy based mainly on the relationship between Neural and Symbol Engines. We also discussed the different cognitive capabilities of these systems in detail. In this section, we provide a general overview of these capabilities and reveal the missing components and important challenges in the field.

Table I summarizes different aspects and capabilities of the proposed categories. The following are some observations related to these capabilities and related challenges:

- Multi-step planning is achieved by all studies except the ones where symbolic policies are transformed into their neural counterparts, losing the symbolic planning capability. While this transformation is done to refine the corresponding policies in the robot's continuous sensorimotor experience, 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 has 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 autonomously generated by the interactive exploration of the robot. Closing this gap, probably by developing custom LLMs, is an important challenge that needs to be addressed.
- Explainability, verifiability, and task monitoring are not strongly addressed by any of these systems simultaneously, which affects their adoption in real life and industry. Hence, strong research programs are needed to fulfill the requirements for general adoption.

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

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	✓✓	✓✓	-	✓	✓	✓✓	-	✓✓	✓	-	✓✓	-	✓✓	✓✓	✓✓	✓✓	✓✓

- Neuro-symbolic robotics generally learn from demonstrations or the robot's own interaction experience. However, reinforcement learning for learning neuro-symbolic structures has not received sufficient attention. Integrating reward-based trial and error and 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.
- Active learning and intrinsic motivation are not exploited sufficiently for neuro-symbolic learning. 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 make use of 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.

VII. CONCLUSION

In this paper, we reviewed the studies in the recently emerging field of Neuro-Symbolic Robotics. We offered a taxonomy of these studies that categorized them based on the role of the Neural and Symbol Engines in the respective robotic architectures and how these engines interplay with each other. While there has been significant effort in non-robotics neuro-symbolic agent architectures [1], [5], [6], [8], [9], [151], [152], we concluded that its robotics counter-part 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. LLMs, especially VLMs, seem to be gaining great popularity in neurosymbolic 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 their processing with verifiable symbolic structures and operations so that their error/hallucinations do not cause unintended harm. One important obstacle that must be overcome is the high energy required to train and operate such systems.

REFERENCES

[1] A. d'Avila Garcez and L. C. Lamb, "Neurosymbolic ai: The 3rd wave," *arXiv e-prints*, pp. arXiv–2012, 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, T. Matsuka, N. Iwahashi, E. Oztop, J. Piater *et al.*, "Symbol emergence in cognitive developmental systems: a survey," *IEEE transactions on Cognitive and Developmental Systems*, vol. 11, no. 4, 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 preprint 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 preprint 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," 2024. [Online]. Available: https://arxiv.org/abs/2405.13245

[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," *arXiv preprint arXiv: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 preprint 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 preprint arXiv:2312.08820*, 2023.

[15] M. Ghallab, A. Howe, C. Knoblock, D. McDermott, A. Ram, M. Veloso, D. Weld, and Wilkins, "Pddl – the planning domain definition language," *Yale Center for Computational Vision and Control Technical Report*, 1998.

[16] K. Mourao, R. Petrick, and M. Steedman, "Using kernel perceptrons to learn action effects for planning," in *Proceedings of the International Conference on Cognitive Systems (CogSys 2008)*, 2008, pp. 45–50.

[17] W. Yuan, C. Paxton, K. Desingh, and D. Fox, "SORNet: Spatial object-centric representations for sequential manipulation," in *Conference on Robot Learning*. PMLR, 2022, pp. 148–157.

[18] T. Migimatsu and J. Bohg, "Grounding predicates through actions," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 3498–3504.

[19] M. Diehl, C. Paxton, and K. Ramirez-Amaro, "Automated generation of robotic planning domains from observations," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 6732–6738.

[20] M. Helmert, "The fast downward planning system," *Journal of Artificial Intelligence Research*, vol. 26, pp. 191–246, 2006.

[21] S. R. Ahmadzadeh, A. Paikan, F. Mastrogiovanni, L. Natale, P. Kormushev, and D. G. Caldwell, "Learning symbolic representations of actions from human demonstrations," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 3801–3808.

[22] 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 *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 5717–5724.

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

[24] V. Sarathy, D. Kasenberg, S. Goel, J. Sinapov, and M. Scheutz, "Spotter: Extending symbolic planning operators through targeted reinforcement learning," *arXiv preprint arXiv:2012.13037*, 2020.

[25] F. S. Lay, A. S. Bauer, A. Albu-Schäffer, F. Stulp, 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.

[26] 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," *Journal of Artificial Intelligence Research*, vol. 75, pp. 709–745, 2022.

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

[28] T. Silver, R. Chitnis, J. Tenenbaum, L. P. Kaelbling, and T. Lozano-Pérez, "Learning symbolic operators for task and motion planning," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 3182–3189.

[29] 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 preprint arXiv:2402.11871*, 2024.

[30] T. Taniguchi, T. Nagai, T. Nakamura, N. Iwahashi, T. Ogata, and H. Asoh, "Symbol emergence in robotics: a survey," *Advanced Robotics*, vol. 30, no. 11-12, pp. 706–728, 2016.

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

[32] T. Nakamura, Y. Ando, T. Nagai, and M. Kaneko, "Concept formation by robots using an infinite mixture of models," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[33] 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.

[34] T. Taniguchi, T. Nakamura, M. Suzuki, R. Kuniyasu, K. Hayashi, 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.

[35] 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.

[36] O. B. Ozturkcu, E. Ugur, and E. Oztop, "High-level representations through unconstrained sensorimotor learning," in *2020 Joint IEEE 10th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*. IEEE, 2020, pp. 1–6.

[37] N. Gopalan, E. Rosen, G. Konidaris, and S. Tellex, "Simultaneously learning transferable symbols and language groundings from perceptual data for instruction following," *Robotics: Science and Systems XVI*, 2020.

[38] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[39] A. Ahmetoglu, B. Celik, E. Oztop, and E. Ugur, "Discovering predictive relational object symbols with symbolic attentive layers," *IEEE Robotics and Automation Letters*, 2024.

[40] A. Ahmetoğlu, E. Öztop, and E. Uğur, "Deep multi-object symbol learning with self-attention based predictors," in *2023 31st Signal Processing and Communications Applications Conference (SIU)*. IEEE, 2023, pp. 1–4.

[41] A. Ahmetoglu, E. Oztop, and E. Ugur, "Symbolic manipulation planning with discovered object and relational predicates," *IEEE Robotics and Automation Letters*, 2025.

[42] M. Cakmak, M. R. Dogar, E. Ugur, and E. Sahin, "Affordances as a framework for robot control," in *7th International Conference on Epigenetic Robotics*, 2007.

[43] 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 *Conference on Robot Learning*. PMLR, 2020, pp. 1439–1456.

[44] E. Ugur, Y. Nagai, E. Sahin, and E. Oztop, "Staged development of robot skills: Behavior formation, affordance learning and imitation with

motionese," *IEEE Transactions on Autonomous Mental Development*, vol. 7, no. 2, pp. 119–139, 2015.

[45] R. Chitnis, T. Silver, J. B. Tenenbaum, T. Lozano-Perez, and L. P. Kaelbling, "Learning neuro-symbolic relational transition models for bilevel planning," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 4166–4173.

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

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

[48] 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 *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4107–4114.

[49] ——, "Learning compositional models of robot skills for task and motion planning," *The International Journal of Robotics Research*, vol. 40, no. 6-7, pp. 866–894, 2021.

[50] C. R. Garrett, T. Lozano-Pérez, and L. P. Kaelbling, "Sample-based methods for factored task and motion planning," in *Robotics: Science and Systems*, 2017.

[51] ——, "PDDLStream: Integrating symbolic planners and blackbox samplers via optimistic adaptive planning," in *Proceedings of the international conference on automated planning and scheduling*, vol. 30, 2020, pp. 440–448.

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

[53] T. Silver, R. Chitnis, N. Kumar, W. McClinton, T. Lozano-Pérez, L. Kaelbling, and J. B. Tenenbaum, "Predicate invention for bilevel planning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 10, 2023, pp. 12 120–12 129.

[54] 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 *7th Annual Conference on Robot Learning*, 2023.

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

[56] 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 preprint arXiv:2312.08566*, 2023.

[57] 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.

[58] G. Konidaris and A. Barto, "Skill discovery in continuous reinforcement learning domains using skill chaining," *Advances in neural information processing systems*, vol. 22, 2009.

[59] 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.

[60] A. Bagaria and G. Konidaris, "Option discovery using deep skill chaining," in *International Conference on Learning Representations*, 2019.

[61] A. Bagaria, J. K. Senthil, and G. Konidaris, "Skill discovery for exploration and planning using deep skill graphs," in *International Conference on Machine Learning*. PMLR, 2021, pp. 521–531.

[62] N. Shah, "Learning neuro-symbolic abstractions for robot planning and learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 21, 2024, pp. 23 417–23 418.

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

[64] D. Molina, K. Kumar, and S. Srivastava, "Learn and link: Learning critical regions for efficient planning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 10 605–10 611.

[65] N. Shah and S. Srivastava, "Using deep learning to bootstrap abstractions for hierarchical robot planning," *arXiv preprint arXiv:2202.00907*, 2022.

[66] 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 preprint arXiv:2410.23156*, 2024.

[67] 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 preprint arXiv:2501.00296*, 2024.

[68] E. Saccon, A. Tikna, D. De Martini, E. Lamon, L. Palopoli, and M. Roveri, "When prolog meets generative models: a new approach for managing knowledge and planning in robotic applications," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 17 065–17 071.

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

[70] 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 preprint arXiv:2405.19758*, 2024.

[71] J. Sun, H. Sun, T. Han, and B. Zhou, "Neuro-symbolic program search for autonomous driving decision module design," in *Conference on Robot Learning*. PMLR, 2021, pp. 21–30.

[72] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," *arXiv preprint arXiv:2204.01691*, 2022.

[73] 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.

[74] G. Tziafas and H. Kasaei, "Enhancing interpretability and interactivity in robot manipulation: A neurosymbolic approach," *arXiv preprint arXiv:2210.00858*, 2022.

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

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

[77] 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.

[78] Y. Zhu, J. Tremblay, S. Birchfield, and Y. Zhu, "Hierarchical planning for long-horizon manipulation with geometric and symbolic scene graphs," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6541–6548.

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

[80] 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," *arXiv preprint arXiv:2006.05398*, 2020.

[81] 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 *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 447–454.

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

[83] L. Lamanna, A. Saetti, L. Serafini, A. Gerevini, P. Traverso *et al.*, "Online learning of action models for pddl planning." in *IJCAI*, 2021, pp. 4112–4118.

[84] 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.

[85] T. Mota and M. Sridharan, "Commonsense reasoning and knowledge acquisition to guide deep learning on robots," in *Robotics: science and systems*, 2019.

[86] 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, no. 5-6, pp. 291–303, 2022.

[87] S. Cheng and D. Xu, "League: Guided skill learning and abstraction for long-horizon manipulation," *IEEE Robotics and Automation Letters*, 2023.

[88] U. Rakhman, J. Ahn, and C. Nam, "Fully automatic data collection for neuro-symbolic task planning for mobile robot navigation," in *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2021, pp. 450–455.

[89] L. Lamanna, L. Serafini, M. Faridghasemnia, A. Saffiotti, A. Saetti, A. Gerevini, and P. Traverso, "Planning for learning object properties," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 10, 2023, pp. 12 005–12 013.

[90] 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.

[91] O. Bastani, Y. Pu, and A. Solar-Lezama, "Verifiable reinforcement learning via policy extraction," *Advances in neural information processing systems*, vol. 31, 2018.

[92] G. Weiss, Y. Goldberg, and E. Yahav, "Extracting automata from recurrent neural networks using queries and counterexamples," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5247–5256.

[93] 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.

[94] S. Carr, N. Jansen, and U. Topcu, "Verifiable rnn-based policies for pomdps under temporal logic constraints," *arXiv preprint arXiv:2002.05615*, 2020.

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

[96] G. Anderson, A. Verma, I. Dillig, and S. Chaudhuri, "Neurosymbolic reinforcement learning with formally verified exploration," *Advances in neural information processing systems*, vol. 33, pp. 6172–6183, 2020.

[97] 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 *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 7973–7980.

[98] 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 *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=rJgMlhRctm

[99] 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," *arXiv preprint arXiv:2212.06817*, 2022.

[100] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," *arXiv preprint arXiv:2307.15818*, 2023.

[101] A. O'Neill, A. Rehman, A. Gupta, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models," *arXiv preprint arXiv:2310.08864*, 2023.

[102] 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 preprint arXiv:2408.10845*, 2024.

[103] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi *et al.*, "Openvla: An open-source vision-language-action model," *arXiv preprint arXiv:2406.09246*, 2024.

[104] S. Dey, J.-N. Zaech, N. Nikolov, L. Van Gool, and D. P. Paudel, "Revla: Reverting visual domain limitation of robotic foundation models," *arXiv preprint arXiv:2409.15250*, 2024.

[105] H. Zhen, X. Qiu, P. Chen, J. Yang, X. Yan, Y. Du, Y. Hong, and C. Gan, "3D-VLA: A 3d vision-language-action generative world model," *arXiv preprint arXiv:2403.09631*, 2024.

[106] K. Kawaharazuka, T. Matsushima, A. Gambardella, J. Guo, C. Paxton, and A. Zeng, "Real-world robot applications of foundation models: a review," *Advanced Robotics*, vol. 38, no. 18, pp. 1232–1254, 2024. [Online]. Available: https://doi.org/10.1080/01691864.2024.2408593

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

[108] 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.

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

[110] M. Zadem, S. Mover, and S. M. Nguyen, "Goal space abstraction in hierarchical reinforcement learning via reachability analysis," *arXiv preprint arXiv:2309.07168*, 2023.

[111] ——, "Reconciling spatial and temporal abstractions for goal representation," *ICLR*, 2024.

[112] M. Zadem, S. Mover *et al.*, "Emergence of a symbolic goal representation with an intelligent tutoring system based on intrinsic motivation," in *Intrinsically-Motivated and Open-Ended Learning Workshop@ NeurIPS2023*, 2023, pp. 423–428.

[113] J. Pisokas and U. Nehmzow, "Experiments in subsymbolic action planning with mobile robots," in *Symposium on Adaptive Agents and Multi-agent Systems*. Springer, 2003, pp. 216–229.

[114] E. Ugur, E. Sahin, and E. Öztop, "Affordance learning from range data for multi-step planning." in *EpiRob*, 2009.

[115] E. Ugur, E. Şahin, and E. Oztop, "Unsupervised learning of object affordances for planning in a mobile manipulation platform," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 4312–4317.

[116] 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.

[117] T. Nakamura, T. Nagai, and N. Iwahashi, "Multimodal categorization by hierarchical dirichlet process," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 1520–1525.

[118] Y. Ando, T. Nakamura, T. Araki, and T. Nagai, "Formation of hierarchical object concept using hierarchical latent Dirichlet allocation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 2272–2279.

[119] T. Nakamura, T. Nagai, and N. Iwahashi, "Bag of multimodal LDA models for concept formation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 6233–6238.

[120] ——, "Bag of multimodal hierarchical dirichlet processes: Model of complex conceptual structure for intelligent robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 3818–3823.

[121] 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 *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 1623–1630.

[122] 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 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014, pp. 600 – 607.

[123] J. Nishihara, T. Nakamura, and T. Nagai, "Online algorithm for robots to learn object concepts and language model," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 9, no. 3, pp. 255–268, 2016.

[124] 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.

[125] 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 Transactions on Cognitive and Developmental Systems*, vol. 8, no. 4, pp. 285–297, 2016.

[126] A. Taniguchi, Y. Hagiwara, T. Taniguchi, and T. Inamura, "Online spatial concept and lexical acquisition with simultaneous localization and mapping," in *2017 ieee/rsj international conference on intelligent robots and systems (IROS)*, 2017, pp. 811–818.

[127] 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.

[128] 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.

[129] 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 *2023 IEEE/SICE International Symposium on System Integration (SII)*, 2023, pp. 1–8.

[130] 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.

[131] G. Konidaris, L. P. Kaelbling, and T. Lozano-Perez, "Symbol acquisition for probabilistic high-level planning," in *AAAI Press/International Joint Conferences on Artificial Intelligence*, 2015.

[132] B. Ames, A. Thackston, and G. Konidaris, "Learning symbolic representations for planning with parameterized skills," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 526–533.

[133] A. Pacheck, S. James, G. Konidaris, and H. Kress-Gazit, "Automatic encoding and repair of reactive high-level tasks with learned ab-

stract representations," *The International Journal of Robotics Research*, vol. 42, no. 4-5, pp. 263–288, 2023.

[134] S. James, B. Rosman, and G. Konidaris, "Learning portable representations for high-level planning," in *International Conference on Machine Learning*. PMLR, 2020, pp. 4682–4691.

[135] ——, "Autonomous learning of object-centric abstractions for high-level planning," in *International Conference on Learning Representations*, 2022.

[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 *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 2627–2633.

[137] ——, "Refining discovered symbols with multi-step interaction experience," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, 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] 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 *Conference on Robot Learning*, 2023, pp. 2278–2287.

[140] 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," *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.

[141] 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," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[142] A. Madaan, N. Tandon, P. Gupta, S. Hallinan, L. Gao, S. Wiegreffe, U. Alon, N. Dziri, S. Prabhumoye, Y. Yang *et al.*, "Self-refine: Iterative refinement with self-feedback," *Advances in Neural Information Processing Systems*, vol. 36, pp. 46 534–46 594, 2023.

[143] N. Shinn, F. Cassano, A. Gopinath, K. Narasimhan, and S. Yao, "Reflexion: Language agents with verbal reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 36, pp. 8634–8652, 2023.

[144] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "React: Synergizing reasoning and acting in language models," in *International Conference on Learning Representations (ICLR)*, 2023.

[145] 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," *Advances in Neural Information Processing Systems*, vol. 36, pp. 68 539–68 551, 2023.

[146] 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.

[147] A. Lazaridou and M. Baroni, "Emergent multi-agent communication in the deep learning era," *arXiv preprint arXiv:2006.02419*, 2020. [Online]. Available: https://arxiv.org/abs/2006.02419

[148] L. Steels, "Modeling the cultural evolution of language," *Physics of Life Reviews*, vol. 8, no. 4, pp. 339–356, 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1571064511001060

[149] C. Zhu, M. Dastani, and S. Wang, "A survey of multi-agent deep reinforcement learning with communication," *Autonomous Agents and Multi-Agent Systems*, vol. 38, no. 1, p. 4, 2024.

[150] 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," 2024. [Online]. Available: https://arxiv.org/abs/2409.02645

[151] 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 *2015 AAAI Spring Symposium Series*, 2015.

[152] P. Hitzler and M. K. Sarker, *Neuro-symbolic artificial intelligence: The state of the art*. IOS press, 2022.