

© 2024 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Renkhoff, Justus, Ke Feng, Marc Meier-Doernberg, Alvaro Velasquez, and Houbing Herbert Song. "A Survey on Verification and Validation, Testing and Evaluations of Neurosymbolic Artificial Intelligence." IEEE Transactions on Artificial Intelligence, 2024, 1–15. <https://doi.org/10.1109/TAI.2024.3351798>.

<https://doi.org/10.1109/TAI.2024.3351798>

Access to this work was provided by the University of Maryland, Baltimore County (UMBC) ScholarWorks@UMBC digital repository on the Maryland Shared Open Access (MD-SOAR) platform.

Please provide feedback

Please support the ScholarWorks@UMBC repository by emailing scholarworks-group@umbc.edu and telling us what having access to this work means to you and why it's important to you. Thank you.

A Survey on Verification and Validation, Testing and Evaluations of Neurosymbolic Artificial Intelligence

Justus Renkhoff *, Ke Feng *, Marc Meier-Doernberg, Alvaro Velasquez, and Houbing Herbert Song, *Fellow, IEEE*

Abstract—Neurosymbolic artificial intelligence (AI) is an emerging branch of AI that combines the strengths of symbolic AI and sub-symbolic AI. Symbolic AI is based on the idea that intelligence can be represented using semantically meaningful symbolic rules and representations, while deep learning (DL), or sometimes called sub-symbolic AI, is based on the idea that intelligence emerges from the collective behavior of artificial neurons that are connected to each other. A major drawback of DL is that it acts as a “black box”, meaning that predictions are difficult to explain, making the testing & evaluation (T&E) and validation & verification (V&V) processes of a system that uses sub-symbolic AI a challenge. Since neurosymbolic AI combines the advantages of both symbolic and sub-symbolic AI, this survey explores how neurosymbolic applications can ease the V&V process. This survey considers two taxonomies of neurosymbolic AI, evaluates them, and analyzes which algorithms are commonly used as the symbolic and sub-symbolic components in current applications. Additionally, an overview of current techniques for the T&E and V&V processes of these components is provided. Furthermore, it is investigated how the symbolic part is used for T&E and V&V purposes in current neurosymbolic applications. Our research shows that neurosymbolic AI has great potential to ease the T&E and V&V processes of sub-symbolic AI by leveraging the possibilities of symbolic AI. Additionally, the applicability of current T&E and V&V methods to neurosymbolic AI is assessed, and how different neurosymbolic architectures can impact these methods is explored. It is found that current T&E and V&V techniques are partly sufficient to test, evaluate, verify, or validate the symbolic and sub-symbolic part of neurosymbolic applications independently, while some of them use approaches where current T&E and V&V methods are not applicable by default, and adjustments or even new approaches are needed. Our research shows that there is great potential in using symbolic AI to test, evaluate, verify, or validate the predictions of a sub-symbolic model, making neurosymbolic AI an interesting research direction for safe, secure, and trustworthy AI.

Impact Statement—Neurosymbolic AI allows the combination

Manuscript received January 31, 2023. This work was supported in part by the U.S. National Science Foundation under Grant No. 2309760 and Grant No. 2317117.

Justus Renkhoff and Houbing Herbert Song are with the Security and Optimization for Networked Globe Laboratory (SONG Lab), Department of Information Systems, University of Maryland, Baltimore County, Baltimore, MD 21250 USA (e-mail: justusr1@umbc.edu; h.song@ieee.org).

Ke Feng is with the Department of Electrical Engineering and Computer Science, Embry-Riddle Aeronautical University, Daytona Beach, FL 32114 USA (e-mail: fengk2@my.erau.edu).

Marc Meier-Doernberg is with the Department of Electrical Engineering and Computer Science, Embry-Riddle Aeronautical University, Daytona Beach, FL 32114 USA (e-mail: meierdom@my.erau.edu).

Alvaro Velasquez is with the Department of Computer Science, University of Colorado, Boulder, CO 80309 USA (e-mail: alvaro.velasquez@colorado.edu).

*Justus Renkhoff and Ke Feng are co-first authors.

of symbolic representations or knowledge with the abstraction capabilities of sub-symbolic AI. This poses new challenges for the AI community, but also offers many new opportunities. As neurosymbolic AI is well suited for safety-critical domains such as autonomous systems, we aim to connect the two fields T&E/V&V and neurosymbolic AI with our survey. Since neurosymbolic AI consists of several components, our research provides an overview of individual aspects regarding the T&E/V&V of these components. Through this, we influence current research in the field of T&E/V&V by highlighting opportunities as well as open challenges that emerge from neurosymbolic AI. Our research demonstrates that by combining symbolic and sub-symbolic AI, it is possible to test, evaluate, verify and validate predictions made by non-transparent sub-symbolic models. Accordingly, we provide an overview of current applications leveraging different architectures and combinations of symbolic and sub-symbolic AI, aiming to either test and evaluate in order to verify and validate predictions or to ease the T&E/V&V processes. In addition, the evaluation of current T&E/V&V methods for their applicability to neurosymbolic applications revealed a need for testing frameworks that focus on neurosymbolic AI. As a result, we provide other researchers with possible directions for future research in the field of T&E/V&V of neurosymbolic AI.

Index Terms—Neurosymbolic AI, Validation, Verification, Evaluation, Testing, Deep Learning, Safety, Security, Trustworthiness

I. INTRODUCTION

NEUROSYMBOLIC artificial intelligence (AI) is an increasingly important trend in machine learning (ML) and has been referred to as the 3rd wave of artificial intelligence [1]. The word “neuro” in its name implies the use of neural networks, especially deep learning (DL), which is sometimes also referred to as sub-symbolic AI. This technique is known for its powerful learning and abstraction ability, allowing models to find underlying patterns in large datasets or learn complex behaviors [2]. On the other hand, “symbolic” refers to symbolic AI. It is based on the idea that intelligence can be represented using symbols like rules based on logic or other representations of knowledge [3]. Neurosymbolic AI combines these two approaches to create a hybrid system that benefits from the reasoning abilities of symbolic AI and the adaptability of sub-symbolic AI, opening new opportunities to improve a variety of different AI branches [4], [5].

A disadvantage of sub-symbolic AI is its nature of being a “black box”. This means that predictions made by these systems can be challenging to explain. Therefore, when an edge case leads to a system failure, it is often hard to find the

reason for it. Accordingly, the rigorous testing & evaluation (T&E) and validation & verification (V&V) of these “black box” is a relevant topic recognized by governments [6] and discussed in current literature [7], [8]. As neurosymbolic systems incorporate a sub-symbolic component, this work aims to provide an overview of current techniques used to validate and verify the symbolic as well as sub-symbolic component, and how the architecture of neurosymbolic systems affects this process and can be used for V&V purposes.

In software engineering, common terms are testing & evaluation or T&E and verification & validation or V&V. As defined by Wallace and Fujii in [9], V&V intends to ensure that software performs as intended and meets certain quality and reliability standards. T&E are the methods and processes used to carry out V&V. Validation refers to the process of ensuring that a system performs as expected and delivers the desired result with sufficient accuracy, while verification focuses on checking if the design and implementation is correct according to the specified requirements [10]. Usually, verification is a process that takes place during development, while validation occurs at the end to evaluate if the program “does what it’s supposed to do” [11]. For reasons of readability, we primarily use the term V&V in the following.

Recent frameworks propose methods to validate and verify symbolic and sub-symbolic AI, but discussing how the architecture of neurosymbolic AI can benefit the V&V process of the system as a whole has not received enough attention yet. Therefore, this paper focuses on two areas. First, the concept of V&V is mapped to symbolic and sub-symbolic AI, and an overview of current techniques and procedures used during the V&V process is provided. Secondly, it assesses how different neurosymbolic applications use the symbolic side to enable V&V of the sub-symbolic component. For this purpose, two different taxonomies of neurosymbolic AI are addressed, which categorize applications based on their architecture. 1) In 2020, Kautz proposed six possible designs of neurosymbolic systems [12]. 2) An alternative taxonomy was introduced by Yu et al. [13] in 2021. These taxonomies are discussed and compared. Based on this, it is analyzed how current neurosymbolic applications leverage these architectures to use the symbolic component to make the sub-symbolic part more transparent, accurate, or safe, therefore enabling the V&V process through a neurosymbolic system design. The structure of the discussion within this paper is visualized in Fig. 1.

Our work demonstrates that some of the current testing methods used for V&V are applicable to neurosymbolic AI. In particular, the combination of knowledge graphs (KGs) and DL is common, and it would be interesting to design a dedicated testing framework based on current techniques to validate neurosymbolic AI as a whole. However, there are also neurosymbolic AI applications that are not easy to test with current means. With this work, we show that there is much research potential in this area, and advocate the awareness of V&V for neurosymbolic AI systems and AI in general. Overall, this paper makes the following contributions:

- Present and compare two current taxonomies of neurosymbolic AI.
- Map the concepts of V&V as used in software engineer-

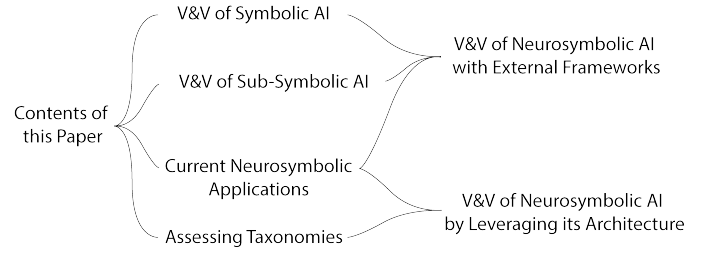


Fig. 1. Contents of this paper.

ing to symbolic and sub-symbolic AI.

- Survey current V&V approaches for symbolic and sub-symbolic AI.
- Analyze the applicability of current V&V methods to neurosymbolic applications.
- Investigate how symbolic AI can support the V&V process of sub-symbolic AI within a neurosymbolic system.
- Discuss opportunities and challenges of V&V in the domain of neurosymbolic AI.

The remainder of this paper is structured as follows: In section II we analyze the related work. After that, in section III we examine and compare two different taxonomies for neurosymbolic AI. Then, in section IV and V we survey the most important methods to verify and validate symbolic AI and sub-symbolic AI respectively. In section VI we analyze if these methods are applicable to current neurosymbolic AI applications and opportunities to leverage different neurosymbolic architectures using the symbolic part to verify and validate sub-symbolic AI. Afterward, in section VII we explain research gaps and problems that might be worth exploring in further research. In section VIII we summarize our findings and explain our planned future work.

II. RELATED WORK

V&V is a crucial process for ensuring the safety and reliability of safety-critical systems. Originally, V&V processes were designed for conventional software without AI components. With the increasing number of modern applications utilizing AI, it becomes crucial to develop approaches for the V&V of systems that use AI as a central element.

A. Surveys on V&V of Machine Learning

V&V of ML is an important topic in current research. For this reason, there are recent works and surveys that deal with this topic [7], [8], [14]–[16]. Current surveys in this domain either deal with a specific area, such as autonomous systems [16], in which ML is used, or only investigate one aspect like ML testing [14], [15] or formal verification of ML [7], which are only parts of the entire V&V process.

In [14], testing of ML is surveyed. The survey presents current testing workflows, the components of an AI-based application that should be tested and provides an overview of properties that require testing as well as the frameworks that can be used to test these properties. Additionally, the survey showcases applications in safety-critical domains that need to

be tested and how the testing workflows and frameworks can be applied to these applications.

Similar, in [15] testing approaches and current testing frameworks are presented. Compared to [14], this survey is not as extensive and does not provide background information about topics like ML in general which is covered in [14], but provides a comprehensive overview of current efforts regarding ML testing.

Huang et al. [8] provide a detailed overview of verification, testing and the interpretability of DL within their survey. They define the terms verification and testing and explain the importance and meaning of properties like the robustness or interpretability of DL. They explain differences between current approaches for V&V of DL and present a variety of testing frameworks and tools to increase the interpretability of DL.

B. Surveys on Neurosymbolic AI

There are multiple recent surveys that cover neurosymbolic AI and its applications in general [17]–[24] and surveys that focus on more specific applications such as graph structures [25], biomedical knowledge graphs [26], or natural language processing [27]. None of the just mentioned surveys covers testing, validation or verification in the domain of neurosymbolic AI and we could not find any surveys covering this topic to this date.

III. TAXONOMIES OF NEUROSYMBOLIC AI

Neurosymbolic AI covers a wide range of applications, and can be implemented in many different ways. This concerns on the one hand the selection of methods used on the symbolic side, and on the other hand how sub-symbolic methods are combined with the symbolic ones. Therefore, it is common to divide neurosymbolic AI into different categories. Accordingly, multiple taxonomies for neurosymbolic AI were proposed [12], [13], [18], [22]. In the following, two current taxonomies are discussed. The one from Kautz [12] and Yu et al. [13] are considered. Both taxonomies categorize neurosymbolic AI based on how the sub-symbolic and symbolic part interact with each other.

A. Kautz’s Taxonomy

Currently, one of the most common categorizations is that of Kautz, who defines six different types of neurosymbolic AI [12]. All of these types represent different system architectures, that try to combine the advantages of symbolic AI with those of sub-symbolic AI. Kautz defines the following categories:

a) Symbolic Neuro symbolic: The input of the system is symbolic, then feed into a Neural Network, which outputs the symbolic result as well. A typical application of Symbolic Neuro Symbolic system is Natural Language Processing (NLP) and has become its Standard Operating Procedure (SOP) [12]. The symbolic input are representing embeddings converted from a combination of words extracted from the original text document. There are a lot of approaches to

perform this conversion, such as word2vec [28], and Glove [29]. Then those symbolic inputs are fed to a neural network that learns the underlying pattern to perform certain tasks, such as translation, semantic classification, and chat robot, etc. The output of the neural network is also symbolic in different forms based on the tasks. For example, the output is a sequence of words for translation tasks or a semantic label for classification tasks.

b) Symbolic[Neuro]: This type of neurosymbolic AI uses a symbolic approach as a problem solver in a neural pattern recognition subroutine. It is currently already being used in many fields. One of the best-known applications is AlphaGo Zero [30]. Kautz states that most current autonomous vehicles and robots utilize this approach, but do not reference any applications from this domain.

c) Neuro|Symbolic: The Neuro|Symbolic system performs symbolic reasoning based on non-symbolic input by leveraging neural networks to transform non-symbolic input (for example images) into a symbolic representation. The outputs of neural networks are fed into a symbolic representation which is used by a symbolic system to perform a complementary task such as query answering [1]. All building blocks are connected so that learning happens in unison. Garcez and Lamb [1] name the neuroßsymbolic concept learner [31] and DeepProbLog [32] as examples.

d) Neuro: Symbolic \rightarrow Neuro: Kautz describes this category as using the SOP which refers to the “Symbolic Neuro symbolic” category. It has a special training regime based on symbolic rules. An example for this method is an application by Lample and Charton [33] which simplifies mathematical expressions. The description of this category is very abstract and Kautz refers to a formula for the training regime that is not explained further, which makes this category rather difficult to grasp.

e) Neuro_{Symbolic}: Within this category, the symbolic part’s purpose is to “transform symbolic rules into templates for structures within the neural network” [12]. Kautz references two examples, which are [34] and [35], to show how this concept can be used to integrate abstraction and part-of hierarchies into neural networks.

f) Neuro[Symbolic]: Neuro[symbolic] is inspired by the “thinking fast and slow” theory from Kahneman [36], who explains that the human brain has two different systems to make decisions. Neural Networks are similar to system 1, which operates automatically by instinct without control. The symbolic part is similar to system 2, which needs attention and effort to operate. Just like a human brain, most of the time system 1 is making decisions until it decides to invoke system 2 is necessary. A Neuro[symbolic] system relies on a neural network, and the embedded symbolic AI assists if invoked by the neural network. This type of neurosymbolic AI is considered to have the highest potential by Kautz [12]. An example is a mouse-maze. Neural Networks recognize this task and invoke the symbolic engine, an algorithm to find the shortest path. The symbolic engine output the path with marks on the map which show the path. Then the neural network has been trained to interpret the marks and follow its guide to find the exit.

Kautz’s categorization demonstrates how the symbolic part cooperates with the sub-symbolic part of the application. This categorization is useful to understand how an application as a whole works, but it also brings some problems with it. His categorization is very fine, and often it is difficult to clearly determine to which category an application belongs. Kautz explains some categories only superficially and gives a few examples, which makes it difficult to understand certain categories thoroughly. While for other categories, there are no applications yet, so it is questionable whether they are at all useful in practice. In addition, the names of his category are not well-chosen. The categories, when pronounced, are sometimes impossible to tell apart and confusion can quickly arise.

B. Yu’s Taxonomy

Because of the critique on Kautz’s survey, we present another survey by Yu et al. [13] which provides an overview of current neurosymbolic applications and presents an alternative taxonomy. In their paper, current neurosymbolic AI applications are studied and divided into three groups: **learning for reasoning**, **reasoning for learning**, and **learning-reasoning**. Just like Kautz’s taxonomy, the categories Yu et al. define represent how the symbolic part interacts with the sub-symbolic part of the application. In the following paragraphs, the taxonomy of Yu et al. will be explained shortly:

1) *Learning for Reasoning*: This approach integrates sub-symbolic processes to enhance symbolic problem solving. Essentially, the sub-symbolic component narrows the search domain for the symbolic solver, optimizing the problem-solving process. This integration is depicted in Fig. 2. Another

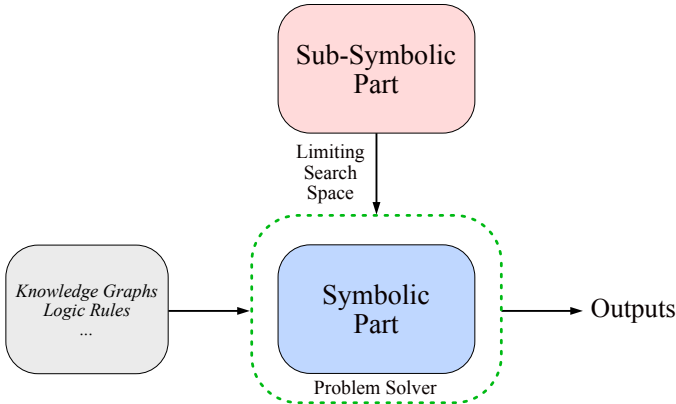


Fig. 2. Flowchart of the Learning for Reasoning type of neurosymbolic AI. The sub-symbolic component is used to limit the search space for the symbolic part. Therefore, it is accelerating the process [13].

way is that the sub-symbolic part converts unstructured data into symbols, to enable efficient symbolic reasoning as shown in Fig. 3.

2) *Reasoning for Learning*: In this model, the roles are reversed: the sub-symbolic element primarily solves problems while the symbolic component supplements the neural network. This support manifests in two ways: firstly, by directing

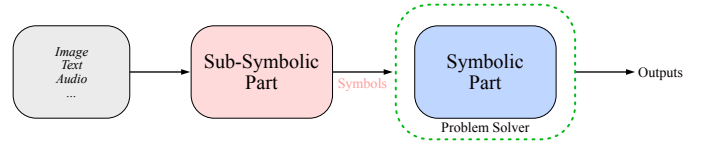


Fig. 3. Flowchart of the Learning for Reasoning type of neurosymbolic AI. In this version of Learning for Reasoning, the sub-symbolic part transforms the knowledge that can be obtained from data to symbols [13].

the neural network during its training phase, and secondly, by imposing constraints during prediction to prevent unsafe outcomes. Fig. 4 illustrates this architecture.

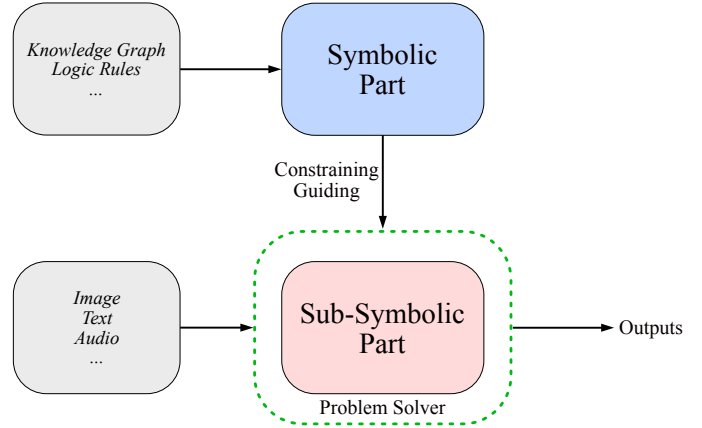


Fig. 4. Flowchart of the Reasoning for Learning type of neurosymbolic AI. Here, the symbolic part can guide or constrain the sub-symbolic part [13].

3) *Learning-Reasoning*: This variant represents a synergistic combination where symbolic and sub-symbolic elements collaborate equally in problem solving. Each component’s output directly informs the other’s input, creating a reciprocal and dynamic interaction. This bidirectional influence is visualized in Fig. 5.

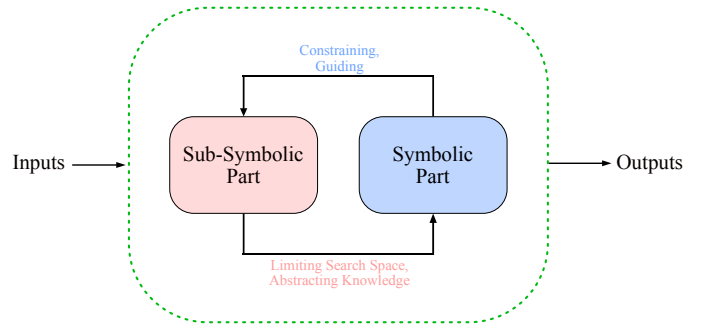


Fig. 5. Flowchart of the Learning-Reasoning type of neurosymbolic AI. Here, the characteristics of the other architectures are combined and the two parts are in constant interaction [13].

4) *Example*: Neurosymbolic approaches that implement safe reinforcement learning via shielding [37] are great examples to showcase this taxonomy as the shielding can be implemented in multiple ways. For a control task, a sub-symbolic model predicts an action, while a so called, safety shield, synthesized from safety specifications specified in temporal logic, ensures that every action is safe. If this application

is implemented following the Learning for Reasoning or Learning-Reasoning design, first, the sub-symbolic part would make a decision based on its inputs from the environment. The decision is then given to the safety shield, that checks if the predicted action is safe and would then make minimal adjustments, if the action is determined to be unsafe. This concept would be categorized as Learning-Reasoning, in case feedback is provided to the sub-symbolic part, letting it know, that the action was replaced or not. If no feedback is provided, it would be Learning for Reasoning. This concept can be seen in Fig. 6.

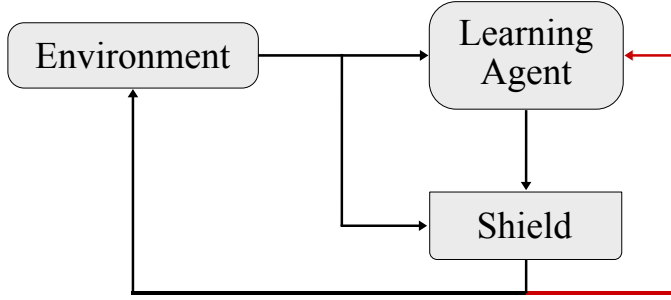


Fig. 6. After the agents (sub-symbolic part) predicts an action based on the inputs from the environment, a safety shield (symbolic part) checks if this decision is safe and replaces it with a safe action if necessary. It is optional (indicated in red) to provide the agent with the information that the action was replaced or not [38].

The work [38] is similar to [37], but extends the paper by presenting an additional architecture in which the shield is inserted before the sub-symbolic part. This allows the shield to limit the action space to make sure that every action the sub-symbolic part can choose from is safe. This design would be “Reasoning for Learning”. This concept can be seen in Fig. 7.

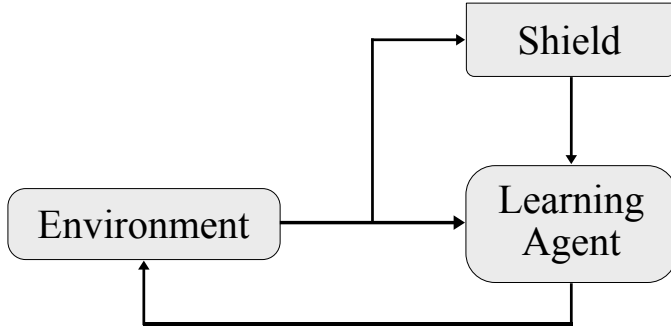


Fig. 7. A safety shield (symbolic part) limits the actions the agent (sub-symbolic part) is able to choose from. Therefore, the agent is only able to choose from a set of safe actions [38].

In their survey, Yu et al. [13] examine a wide range of current applications and classify them. They show that a variety of symbolic techniques can appear in every category of neurosymbolic AI. For example, first-order logic is used as a symbolic method in applications of all categories. This shows that the selection of the algorithms and methods for the symbolic as well as the sub-symbolic part is independent of the associated category. The categories in Yu’s taxonomy are only based on the interaction of the symbolic and sub-symbolic

component. In the following sections, we will analyze which frameworks and methods are currently used to test the most common symbolic and sub-symbolic methods and how the symbolic part of the application can contribute to the testing of the sub-symbolic component.

IV. V&V OF SYMBOLIC AI

As a first step, the V&V process of the two components of a neurosymbolic application are considered independently, with this section focusing on the symbolic component. Yu et al. [13] shows that three methods in particular are used frequently as the symbolic part of a neurosymbolic AI system. These are propositional logic, first-order logic, and KGs. In the following section, the concepts of V&V are mapped to these techniques and the capabilities to validate and verify of symbolic AI are assessed.

A. Mapping V&V to Logical Systems

In term of V&V, the following properties are the most relevant: 1) Validity: A formula is valid if it is true under every possible interpretation or assignment. In logic, this means that if all premises of a statement are true, it is impossible that the conclusion is false. 2) Soundness: A logical system is sound if every statement that can be derived from the systems is true and an argument is sound if it is valid and its premises are true.

When mapping logical arguments to the V&V process, the validity of a logic argument can be analogized to the verification phase. This is because a valid logical argument ensures structural correctness given that all premises are true, though it doesn’t necessarily affirm the truth of those premises. This is similar to the verification process checking if the implementation or design of a system is correct according to its specification. On the other hand, soundness aligns with the validation phase, as an argument is deemed sound only when all its premises are unequivocally true. This is analog to the validation phase, as this ensures that the output of a system is as expected and correct.

B. Verification of Logic

If a general algorithm can be found to prove the validity (true/false) of a logic argument, it is called decidable. Therefore, the question is: Are propositional logic and first-order logic decidable?

1) Propositional logic is decidable. The validity of a statement can be determined by a truth table. Truth tables are a fundamental process of computer science. As Anellis’s research shows, it appears that this technique was used as early as the 19th century [39]. The complexity of this proof grows exponentially with the number of variables. Therefore, truth tables are in practice only usable for statements with a small number of propositional variables. Semantic tableau also called the truth tree method is an elegant alternative to truth tables [40]. Accordingly, it is possible to validate the symbolic part of a neurosymbolic AI application that uses propositional logic using this technique, even if current standards are rather inefficient.

2) There is no general algorithm to check the validity of a first-order logic statement. Therefore, first-order logic statements are undecidable. However, this does not mean that it is impossible to show the validity of individual statements. Truth trees can be used to show the validity of first-order logic statements, but if the statement is invalid, the algorithm will run infinitely. To show an invalid statement, a countermodel has to be found.

As described above, algorithms have been found to check the validity of logic arguments. Even though first-order logic is not decidable, tools like [41], can be used to verify it in many cases. However, validity does not depend on whether the premises are true. This means that the following statement would be valid in terms of logic:

All animals are birds.

All dogs are animals.

Therefore, all dogs are birds.

The above example shows that a statement can be valid but not sound. Soundness, as explained before, describes that not only the syntax but also the semantic is valid. Therefore, additional knowledge has to be used, to validate the semantic of such an argument. For this purpose, KGs could be used for validation purposes, which again have to be validated, too. This problem is considered in more detail in the following section.

C. Validation of Knowledge Graphs

KGs are an increasingly important component of current applications. Accordingly, there are numerous methods for validating these graphs. The survey “Knowledge Graph Validation” by Huaman et al. gives an overview of current methods and tools [42]. Within this survey, several typical error sources which frequently occur in KG are described, and an overview is provided of which current tools are able to detect and correct such errors in order to create the most valid KG possible. A broad variety of tools are available to validate KGs.

1) *Corroborative Fact Validation (COPAAL)* [43]: To validate KGs or semantic statements, COPAAL computes a so-called mutual information (MI) score. The method tries to find alternative sources on the web to validate a statement. The paper gives an example of how this method works; a given statement could be: “Barack Obama is a US citizen”. Using open databases and KGs such as DBpedia 2016-10¹, the method looks for similar statements that imply or refute that Barack Obama is a US citizen. E.g. the data could show that his place of birth is in the USA which would make it highly likely that he is a US citizen or if the method finds a source that states that he was a US President, it confirms that the original statement is very likely to be correct, giving it a high MI score.

2) *Deep Fact Validation (DeFacto)* [44]: To validate knowledge, DeFacto is an algorithm that tries to find supporting information about a given fact in the information as well as supporting information from trustworthy sources. Additionally, it provides a score that represents the confidence DeFacto has when assessing the validity of a fact.

3) *Temporal Information Scoping (TISCO)* [45]: TISCO adds another component. This procedure tries to assign times to facts, since many assertions are only true at certain times. E.g. athletes regularly change their clubs, people may have different professions or live in different places at different points in their lives. Therefore, it is important not only to validate the facts, but also to link them to points in time in order to establish a timeline.

In addition to the above-mentioned procedures, there are other similar procedures with the same goal. In all procedures, different databases or the web are searched based on an assertion in order to confirm and validate statements. Other popular methods are FactCheck [46], FacTify [47], Leopard [48], Surface [49] and S3K [50]. Furthermore, there are already well build KGs available that are tested and highly validated, like YAGO [51] or Conceptnet [52]. YAGO is used by IBM in their Watson artificial intelligence system [53] and stores knowledge about people, cities, countries, movies, and organizations. It was build with data from Wikipedia², WordNet [54], which is also a widely used KG, and GeoNames³. ConceptNet is a knowledge graph that links words and phrases with labeled edges. The information comes from a variety of sources, including crowdsourcing, expert-generated material, and games. Another popular KG is DBpedia⁴ which builds on knowledge from Wikipedia documents.

V. V&V OF SUB-SYMBOLIC AI

V&V in the context of sub-symbolic AI is an exciting topic and also a big challenge, as deep learning is also often referred to as a “black box” and is rather opaque in its decision-making. Accordingly, it is a challenge to validate and verify the behavior of these systems. In order to verify a system, it is usually checked whether certain requirements are met. This is usually done with the help of formal methods and is a current challenge for systems using sub-symbolic AI due to its complexity. For the validation of sub-symbolic AI different testing methods are used to check different properties like the correctness or robustness of a system.

A. Verification of Sub-Symbolic AI

In the survey of Huang et al. [8] various applications to verify sub-symbolic AI are presented. They provide a taxonomy for different verification approaches and define multiple properties that can be verified. In the following these approaches and properties as defined in [8] are summarized.

1) Properties to Verify:

a) *Robustness*: Robustness can be defined as the ability of a model to make a correct decision even in situation when the input is noisy or manipulated [55].

²<https://www.wikipedia.org/>, accessed: 01/29/2023

³<https://www.geonames.org/>, accessed: 01/29/2023

⁴<https://www.dbpedia.org/resources/knowledge-graphs/>, accessed: 01/26/2023

¹https://www.faa.gov/air_traffic/publications/atpubs/atc_html/chap5_section_7.html, accessed: 01/26/2023

b) Reachability & Interval: The reachability and interval are two very similar properties, closely connected to each other. Verifying the reachability means, that for a certain input the highest possible and lowest possible output is verified. Verifying the interval is very similar, as it is an over-approximation of the reachability.

c) Lipschitzian: This property describes how the output changes when small changes are made to the input. When verifying this property, the change in output should remain below a specified distance.

2) *Approaches:*

a) Search-Based: Verification algorithms belonging to this type verify the system through exhaustive searching. This approach uses algorithms such as the Monte-Carlo Tree Search for verification purposes [56].

b) Constraint Solving: Algorithms that leverage this approach convert neural network into constraints which are easier to verify because they are no longer a “black box”. For the verification of the resulting constraints, solvers like the SAT solver can be used.

c) Over-Approximation: Here, an over-approximation of possible outputs for an input is calculated for verification purposes.

d) Global Optimization: As the name suggests, these approaches are based on global optimization techniques. An example for this is the tool DeepGo [57] that uses global optimization techniques for verification in respect to reachability and robustness properties.

An in-depth explanation for the verification of sub-symbolic AI can be found in [8]. Table I provides an overview of current approaches that can be used for the verification of sub-symbolic AI.

TABLE I
APPROACHES TO VERIFY SUB-SYMBOLIC AI AS SURVEYED IN [8].

Approach	Publications
Search-Based	[56], [58]
Constraint Solving	[59]–[66]
Over-Approximation	[57], [67]–[70]
Search-Based & Constraint Solving	[71], [72]
Over-Approximation & Constraint Solving	[73]–[75]
Global Optimization	[57], [70]

B. Validation of Sub-Symbolic AI

To validate sub-symbolic AI, a variety of measures can be tested. In [14] these measures as well as the tools and frameworks to test these in order to validate such a system is surveyed. The measures addressed in this survey are the correctness, model relevance, efficiency, fairness, interpretability, privacy and robustness of the system. In [8] especially testing the robustness and increasing the interpretability are addressed. Depending on the use case of the application, some of these properties are particularly important. Within this survey especially the correctness, robustness and interpretability

TABLE II
WORKS ON TESTING THE CORRECTNESS AS SURVEYED IN [14]

Testing Correctness	Publications
Testing Tools	[79], [80]
Testing the Input and Oracle Design	[81]–[86]
Searching Data Bugs	[87], [88]

are considered for validation purposes. In the following, a brief overview of these measures and recent frameworks is provided.

1) *Properties to Validate:*

a) Correctness: Correctness is a fundamental property of a system, representing the probability that it completes a task correctly. Popular methods to measure the correctness are k-fold cross-validation [76] and Bootstrapping [77]. For classification tasks, metrics like accuracy, precision/recall, and ROC Curve are commonly used to measure the correctness. Suitability varies depending on the situation and data balance. Detailed examples can be found in Japkowicz’s workshop [78]. Regression problems can be evaluated using error measurements, such as Mean-Squared-Error (MSE) or Root Mean-Squared-Error (RMSE), which provide insights into expected deviations from the system’s predictions. In summary, choosing the appropriate measurement is crucial to assess the correctness of a sub-symbolic system and should be carefully considered based on the task and data distribution. Table II shows a selection of works that focus on testing the correctness of sub-symbolic AI. It is based on applications surveyed in [14].

b) Robustness: The robustness property itself is similar to the one described in section V-A1a. The difference is that robustness can not only be verified, but also tested and therefore validated. The most common approach to test the robustness is to generate adversarial examples or inputs. Frameworks such as DeepXplore [89], DeepHunter [90] or DLFuzz [91] use adversarial attacks to trigger misbehavior and therefore to test the robustness of a neural network. Techniques such as testing the code coverage, known from conventional software testing, can be adapted to sub-symbolic AI. These approaches maximize a metric called neuron coverage to improve the robustness of a sub-symbolic model. Another approach is to detect adversarial noise that might cause wrong predictions [92], [93]. While these methods focus on images, there are other approaches that focus on generating and detecting adversarial attacks for natural language processing [94] or cybersecurity [95], which can be used to test and improve the robustness of these models.

2) *Interpretability:* Neural networks are often considered to be “black boxes”, because it is a challenge to comprehend the decision-making process of a trained model. However, in safety-critical and ethically sensitive domains, it is crucial to understand this process to prevent discrimination or system failures. Although there is no uniform definition of interpretability, previous work suggests that it refers to the degree to which humans can comprehend the reasoning and logic behind a deep learning system’s decisions [14], [96].

To evaluate interpretability, there are three main categories: Manual assessment, automatic assessment, and evaluation of

interpretability improvement [14]. Manual assessment involves humans in the loop and is evaluated in real applications. Automatic assessment, on the other hand, utilizes proxies to eliminate the need for human involvement. Identifying influential instances belongs to this approach, which can be achieved through two methods: Deletion Diagnostics and Influence Functions [97]. Both methods detect influential instances by measuring the influence of the change to the model when modifying the data sets: Deletion Diagnostics remove data points, while Influence Functions up-weight instances by differentiating the loss function with respect to its parameters. Notable measures of Deletion Diagnostics are DFBETA [98] and Cook’s distance [99].

VI. OPPORTUNITIES

As shown in Fig. 8, one solution is to verify and validate both sides of a neurosymbolic AI separately. Another solution is to leverage the characteristics of the symbolic AI to verify and validate sub-symbolic part. In the following, we will consider both approaches and assess whether and how current testing and validation methods can be applied to the isolated parts of a neurosymbolic application and how current applications leverage the characteristics of symbolic policies to validate or improve the properties of the sub-symbolic part.

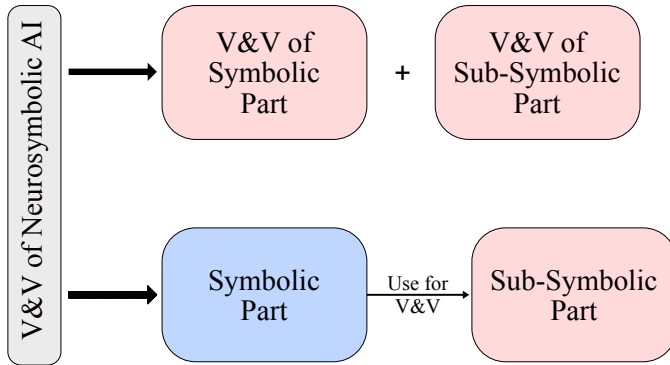


Fig. 8. It is possible to either verify and validate the parts of a neurosymbolic application independently or the symbolic part can be used to either ease or conduct the V&V process of the sub-symbolic part.

A. Using Neurosymbolic System Architectures for V&V

Each of the three different categories of neurosymbolic AI defined by Yu et al. [13] presented in their paper can affect the V&V process differently. For example, in “Reasoning for Learning”, the symbolic part can support the sub-symbolic AI by providing guidelines and constraints through e.g. logic rules. This means that the input is directly applied to the sub-symbolic AI, as shown in Fig. 4. The symbolic part can therefore improve the robustness and correctness of the system by checking, constraining or replacing decisions made by the sub-symbolic model. The category “Learning for Reasoning” uses the symbolic part as problem solver. This means that the inputs directly go into the sub-symbolic part. It is feasible to transform the inputs to the sub-symbolic part to symbolic rules that allow to make transparent decisions and therefore increase the interpretability of the overall system. Both “Learning

for Reasoning” as well as “Reasoning for Learning”, have the potential to improve the efficiency by accelerating the learning process either through guidance by symbolic rules or by limiting the search space with the sub-symbolic model. All of these concepts can also be applied to the category “Learning-Reasoning”. In the following, we give multiple examples based on current neurosymbolic applications that leverage these architectures and explain the opportunities these techniques provide to increase the safety and trustworthiness in AI and especially DL. An overview of selected applications is given in table III.

1) *Safe Reinforcement Learning*: A popular application for neurosymbolic AI is safe reinforcement learning for autonomous control tasks. Alshiekh et al. [37] propose a concept to synthesize a safety shield from formal specifications represented in linear temporal logic. As already mentioned in section III, the integration of this shield into the neurosymbolic systems is very versatile and every architecture according to Yu’s taxonomy is possible. In recent years, several similar approaches have been proposed, often using the “Learning for Reasoning” or “Learning-Reasoning” architecture to make the minimal needed adjustments to guarantee safe actions. One of the more recent works is “Neurosymbolic Reinforcement Learning with Formally Verified Exploration” [100]. The paper introduces a reinforcement learning framework called REVEL. Similar to [37], symbolic rules are used as a verification step within the deep reinforcement learning loop providing a safety shield that keeps the agent from executing unsafe actions. Therefore, this application can be categorized as “Learning-Reasoning”, showing how architecture can help to improve the correctness and robustness of a system. The paper demonstrates the results using a total of 10 benchmarks and compares them with similar state-of-the-art approaches. Compared to Deep Deterministic Policy Gradients (DDPG) [101], the framework performs better in 7 out of 10 scenarios. Compared to Constrained policy optimization (CPO) [102], however, it performs better in only 4 out of 10 cases. The survey “A Review of Safe Reinforcement Learning: Methods, Theory and Applications” by Gu et al. [103] provides an overview of safe reinforcement learning with many different approaches often using neurosymbolic AI for verification purposes. Additionally, the authors maintain a GitHub repository⁵ listing current works in this domain.

2) *Verifiable Reinforcement Learning via Policy Extraction* [104]: Another approach to increase the interpretability and transparency of the decisions of the sub-symbolic part of a neurosymbolic system is to derive rules from predictions. The neurosymbolic framework VIPER, which follows the “Learning for Reasoning” architecture, is doing this by deriving rules from predictions of a neural network. These rules are represented by a decision tree. This approach helps to make decisions easier and more efficient to validate and verify. Furthermore, it makes the decisions of the entire system more transparent.

⁵<https://github.com/chauncygu/Safe-Reinforcement-Learning-Baselines>, accessed: 12/09/2023

TABLE III
A SELECTION OF PAPERS THAT USE SYMBOLIC AI FOR V&V OF THE
SUB-SYMBOLIC PART

Paper	Category	Summary	V&V Aspect
[100]	Learning-Reasoning	Verify predictions using a symbolic “safety policy”	Increasing correctness and robustness by ensuring a safe output
[105]	Non Applicable	Convert 3D objects to code which is then converted to 3D shapes	Improves validity by increasing the interpretability
[104]	Learning for Reasoning	Learn a provable decision-tree policy	Improves validity by increasing the interpretability
[106]	Learning for Reasoning	Learn programmatic policies from tasks that can be described by Markov Decision Processes	Improves validity by increasing the interpretability
[38]	Learning for Reasoning or Learning-Reasoning (based on implementation)	Restrict a DP model or overwrite its decision to allow safe reinforcement learning	Validate predictions or restrict DL model to improve correctness and robustness
[107]	Learning-Reasoning	Learning semantic video representations in a neurosymbolic weak supervised learning setup	Verify learning results by checking against logical specifications

3) *Learning to Synthesize Programs as Interpretable and Generalizable Policies* [106]: This framework follows a similar approach as “Verifiable Reinforcement Learning via Policy Extraction” [104]. The difference is that [106] does not use limited policy representations in the context of decision trees, but learns to synthesize a program solely on rewards. The derived policies can make the decisions more transparent than those of conventional DL methods.

4) *Learning to Infer and Execute 3D Shape Programs* [105]: This application is interesting because unlike the others, it does not quite fit Yu’s taxonomy [13] because it uses a total of two sub-symbolic parts and one symbolic part. Again, the symbolic part gives more accurate results and especially increases transparency compared to existing DL methods. The goal of the program is to represent a 3D object as 3D shapes. For this, first, an object is represented as code by means of a “Neural Program Generator”. Then a “Neural Program Executor” converts the code to 3D shapes. The code is human-readable, and therefore it is possible to see which shapes of the 3D object have been recognized. This increases the interpretability.

5) *LASER* [107]: Huang et al. [107] present a weakly supervised neurosymbolic learning approach to learn semantic video representations. The approach receives videos and spatio-temporal specifications in the form of linear temporal logic (LTL) as inputs. During the learning process an “alignment score” of the specifications and the learned semantic representation is calculated. This allows for a verification of the learned representation. The alignment is optimized during

the learning process.

B. Assessing the Applicability of Current T&E/V&V Methods to Neurosymbolic AI

In this section, we address opportunities we have through current V&V methods to determine where these approaches reach their limits in neurosymbolic applications.

1) *Zero-shot Recognition via Semantic Embeddings and Knowledge Graphs* [108]: This method, which according to Yu’s taxonomy [13] belongs to the category “Reasoning for Learning”, deals with zero-shot learning. The approach deals with unknown classes by using knowledge about previously learned classes and additional semantic embeddings. It uses both semantic embeddings and categorical relationships to predict the classes of unknown pictures. The core of the application consists of two components: One component is a knowledge graph (KG), and the other is a graph convolutional network (GCN). The paper uses multiple configurations of datasets for its experiments. In the first one, the KG is based on relationships from Never-Ending Language Learning (NELL) [109] and images are taken from the Never-Ending Image Learning (NEIL) [110] dataset. In the second configuration, the KG is based on the WordNet [54] database while the images for the GCN model’s training are taken from the ImageNet [111] dataset. The KG can be validated with the previously analyzed methods. The paper investigates how the method behaves when noise is introduced in the KG and when it is completely random. It is shown that the method is quite robust even when noise is present in the KG. However, if it is random, then the outputs are almost random guesses. Therefore, while it is important that the KG is validated by the GCN, which is the problem solver in this procedure, we compensate for noise but do not need to validate the KG perfectly and focus on the GCN. There are several types of the still rather new GCN. The type used in the paper is based on convolutional neural networks (CNNs). This would mean that approaches such as [112] to find robustness guarantees in GCNs could be used for verification and benchmarking tools like [113], [114] for validation purposes. Even though there are some works regarding V&V of GCN it is a rather unexplored topic, which would be exciting to further investigate.

2) *Alpha Go Zero* [30]: AlphaGo Zero is an application developed by DeepMind. It is able to beat the world’s best players in games like Chess or Go. This application is not listed in Yu’s [13] survey, but Kautz’s [12] uses it as an example for his category Symbolic[Neuro]. If this method were included in Yu’s taxonomy, it would belong to the category “Learning for Reasoning”. A neural network evaluates the state of the game on the sub-symbolic part of the application, while a Monte Carlo Tree Search [115] tries to find the optimal move for the given situation on the symbolic part. Therefore, this application has a symbolic problem solver with a neural network supporting the decision-making process. AlphaGo Zero is trained by playing against itself in an attempt to find better moves and thus better models. AlphaGo Zero’s model trains itself and no human-generated data set is needed. This means that the system does not need to be protected against

noisy or manipulated data. In addition, it is not a safety-relevant application. Therefore, robustness is not necessarily in the foreground, since targeted manipulations would be unlikely and futile. While the interpretability of AlphaGo Zero’s decisions is interesting, it is not the priority when testing or verifying the system. The goal of AlphaGo Zero is to develop the strongest possible chess engine that can defeat any opponent. For this reason, the main focus in testing the program is on the correctness. To validate the Monte Carlo Tree Search, all possible moves for each possible game state would have to be evaluated to find the optimal solution. For games like TicTakToe, this would not be a problem, but since games like Chess or Go have too many different game states, this would not be feasible with current technology. To simplify this, the neural network looks at each game situation and evaluates it. Since numerous game states are very similar and similar moves would be optimal, the DL part tries to identify these relationships between the different situations to reduce the possibilities that need to be evaluated. This means that in order for the symbolic problem solver to be able to make the correct decision, the sub-symbolic part must have assessed the situation correctly beforehand. Thus, a labeled test data set would need to be created against which the model could be tested to make sure the game states are detected correctly. However, since AlphaGo Zero has never been beaten by a human, it is impossible to decide whether the human made a mistake in labeling or if AlphaGo Zero made a mistake in evaluating a game situation if differences occur. However, as improvements are constantly being made, especially in the efficiency of this process, it is clear that even though the models so far are very good, there is still room for improvement. The only way to test AlphaGo Zero at the moment is to let it play more games against itself to find better models, even if this is very inefficient. For this reason, however, optimizing the efficiency of AlphaGo Zero’s training is also interesting, since the more efficient this is, the faster better results can be obtained and thus the most important property in this scenario, correctness, is also improved. To summarize this; there is no current test framework that would be applicable to an application like AlphaGo Zero.

3) *DeepProbLog* [116]: DeepProbLog is a neurosymbolic AI framework belonging to the category learning for reasoning according to Yu’s taxonomy. The sub-symbolic part is responsible for the low-level perception task, and the symbolic part then uses the learning result to perform logical inference. In their research, three sets of a total of six experiments are conducted to demonstrate the different abilities of DeepProbLog. In five out of six experiments, DeepProbLog outperforms the DL model itself, showing better generalization ability, less computational complexity and training time, and higher sample efficiency. The tasks in the experiments are the addition of single digits and multi-digits, sorting a list of numbers, and the coin-ball problem [117], where the sub-symbolic part is used to recognize the numbers or colors in an image, and the symbolic part uses the classification results to complete the addition operation or to calculate the probability distribution. The sub-symbolic components used in these tasks are convolutional neural networks (CNNs) with

basic architectures. The experiments used the MNIST data set. Input testing could be conducted to expose robustness flaws [14]. Testing frameworks like DLFuzz could be used to generate adversarial samples and improve the robustness of the CNNs [118]. For the sorting task, the sub-symbolic part uses recurrent neural networks (RNNs) which are similar as the ones used in the work of Bosnjak et al. [119]. These could be tested by TensorFuzz [120], which is used to find undesired behaviors of RNNs. Also, cross-validation could be used during the training to validate the models performance. The symbolic part of DeepProbLog follows the inference process of ProbLog: First, generate the ground instances the query is based on; Second, rewrite the ground logic into a propositional logic formula; Next, the formula is compiled into a Sentential Decision Diagram (SDD) [121] for more efficient evaluation; Finally, calculate the probability. Since this system is based on propositional logic, the symbolic part is decidable and could be verified e.g. using semantic tableau.

VII. OPEN CHALLENGES

Examining the current state of neurosymbolic AI and current V&V methods, we have revealed numerous open challenges. These open challenges address neurosymbolic AI and its applications in general, as well as the V&V methods for both symbolic and sub-symbolic AI.

a) *Investigating New Neurosymbolic Architectures*: The term “neurosymbolic AI” is still relatively new at the time this paper was written. As our research has shown, it is difficult to find papers on the topic on the well-known platforms of ACM and IEEE. However, this is not because no one uses this concept, but because the term is not yet widely used in the scientific community. The works by Kautz [12] and Yu et al. [13] make important contributions by identifying and categorizing existing applications that use this technique. Similar works are published frequently, but there is still no widespread differentiation of different categories of neurosymbolic AI and terms as well as clear definitions must be established in the future. We have criticized Kautz’s taxonomy for the fact that some of his categories are only theoretical with no applications implementing them and thus some categories are not practical at the moment. But this also shows that there are many opportunities to combine symbolic with sub-symbolic AI that have not been explored yet and are worth exploring to find out what potential neurosymbolic AI has.

b) *Efficient Verification of Logic Rules*: Traditional methods like truth tables, which can be used to verify propositional logic, are very computationally intensive as their run-time depends on the number of parameters. This means that these methods do not scale well. However, depending on how the sub-symbolic part is related to the symbolic part, it may not be necessary to fully verify the symbolic part. Neural networks have the advantage that they can usually deal well with noise in the data. That means, if the problem solver is the sub-symbolic part and the symbolic part has only a supporting function, it would be sufficient to approximate a complete verification. This approach could be further investigated and used to balance the computational cost and scalability with the need for accuracy and logical correctness.

c) *Testing of Emerging DL Architectures:* Methods for testing the correctness, robustness, and other metrics for neural networks are well-researched and are constantly being further developed. It happens again and again that new designs for neural networks are developed. These new architectures require either new testing methods or the adapting of existing ones. In the paper “Zero-shot Recognition via Semantic Embedding and Knowledge Graphs” [108] a GCN is used on the sub-symbolic part of the application. It would be interesting to explore whether it is possible to apply methods such as DLFuzz [91] here.

d) *Comparing the Efficiency of Neurosymbolic AI with Comparable Conventional Deep Learning Approaches:* Through neurosymbolic AI it is possible to perform the training process of a DL model in a more targeted way, since the symbolic part can guide and thereby support the sub-symbolic part during training and the decision-making process. Therefore, it would be interesting to compare whether neurosymbolic AI applications are more efficient in terms of runtime and possibly also in terms of energy consumption. Measuring the efficiency of software systems and AI are exciting topics that are currently being researched. Since energy-efficient training AI can save costs for companies and research institutions as well as protect the environment, it is exciting to look at the influence of neurosymbolic AI on the efficiency of training. The assessments could be based on existing metrics and test procedures for evaluating the resource efficiency of ML [122], [123].

e) *Apply Current V&V Methods to Common Neurosymbolic Applications:* It could be tested whether existing V&V methods can be applied to common neurosymbolic applications as explained in the opportunities area. The currently most popular neurosymbolic AI applications could be used as examples. This could be extended and a testing framework for neurosymbolic AI applications could be developed, because there are some configurations that are frequently used. For example, KGs are often combined with CNNs. Test frameworks could be developed for these standard configurations with respect to the architecture of the application.

f) *Development of Dedicated Testing Frameworks for Applications using Neurosymbolic AI:* At present, there are only a few frameworks for testing neurosymbolic applications, as this is still a very new field. While our paper focuses on testing the individual components and using symbolic AI to test the sub-symbolic component within the system, there are first frameworks that test the whole neurosymbolic system as such. These testing frameworks are showing initial success in domain-specific applications. For example, Large Language Models (LLMs) are a popular area of application for neurosymbolic AI. Accordingly, the paper [124] introduces a “diversity measure” based on entropy, Gini impurity, and centroid distance as a metric to determine the probability of failure of LLMs. Furthermore, for the neurosymbolic LASER [107] approach for learning semantic representations of videos a new model checker was needed. Accordingly, they implemented a model checker based on Scallop [125] for verification purposes. This shows that there is a great need for new model checkers and testing procedures for applications

based on neurosymbolic AI.

VIII. CONCLUSION

Within this paper, the current state of neurosymbolic AI was investigated, as well as the current possibilities to test, evaluate verify and validate neurosymbolic AI. Two taxonomies that categorize neurosymbolic applications based on the system’s architecture describing how the symbolic and sub-symbolic parts of the application interact with each other were assessed. Afterwards, the standard procedures to verify and validate common approaches used on the symbolic as well as the sub-symbolic part were surveyed. Based on this, it was analyzed whether it is possible to apply these strategies to popular neurosymbolic applications mentioned in recent surveys. It was found that the applicability of current testing methods strongly relates to the algorithms used on the symbolic and sub-symbolic parts. While there are V&V methods for most approaches used on the symbolic part, these are sometimes too computationally expensive for large-scale projects. Therefore, it is important to question how thorough the testing on this side has to be, since neural networks can handle noisy data well if the symbolic part is only supporting. For the sub-symbolic side, current testing frameworks can often be used for the V&V. These may be modified if necessary, however, this area is still a vivid research area, and it may happen that neurosymbolic applications use concepts for which no current testing framework exist. Furthermore, some applications demonstrate how the symbolic part of the application can be used to make neural networks more transparent, robust or accurate. This approach offers many opportunities and is still very unexplored, so it is exciting to explore this technique in future works including different environments and applications. In addition, it was found that there is a growing need for dedicated testing frameworks specialized for domain-specific neurosymbolic applications.

REFERENCES

- [1] A. d. Garcez and L. C. Lamb, “Neurosymbolic ai: The 3 rd wave,” *Artificial Intelligence Review*, pp. 1–20, 2023.
- [2] C. M. Bishop, “Neural networks and their applications,” *Review of scientific instruments*, vol. 65, no. 6, pp. 1803–1832, 1994.
- [3] M. Garnelo and M. Shanahan, “Reconciling deep learning with symbolic artificial intelligence: representing objects and relations,” *Current Opinion in Behavioral Sciences*, vol. 29, pp. 17–23, 2019.
- [4] K. Acharya, W. Raza, C. Dourado, A. Velasquez, and H. H. Song, “Neurosymbolic reinforcement learning and planning: A survey,” *IEEE Transactions on Artificial Intelligence*, pp. 1–14, 2023.
- [5] A. Velasquez, “Transfer from imprecise and abstract models to autonomous technologies (tiamat),” *Defense Advanced Research Projects Agency (DARPA) Program Solicitation*, 2023.
- [6] Oct 2023. [Online]. Available: <https://www.whitehouse.gov/briefing-room/presidential-actions/2023/10/30/executive-order-on-the-safe-security-and-trustworthy-development-and-use-of-artificial-intelligence/>
- [7] X. Huang, “Safety and reliability of deep learning: (brief overview),” in *Proceedings of the 1st International Workshop on Verification of Autonomous & Robotic Systems*, ser. VARS ’21. New York, NY, USA: Association for Computing Machinery, 2021. [Online]. Available: <https://doi.org/10.1145/3459086.3459636>
- [8] X. Huang, D. Kroening, W. Ruan, J. Sharp, Y. Sun, E. Thamo, M. Wu, and X. Yi, “A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability,” *Computer Science Review*, vol. 37, p. 100270, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1574013719302527>

- [9] D. Wallace and R. Fujii, "Software verification and validation: Its role in computer assurance and its relationship with software project management standards," 1989-09-05 1989. [Online]. Available: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=905731
- [10] R. G. Sargent, "A tutorial on validation and verification of simulation models," in *Proceedings of the 20th conference on Winter simulation*, 1988, pp. 33–39.
- [11] D. Wallace and R. Fujii, "Software verification and validation: an overview," *IEEE Software*, vol. 6, no. 3, pp. 10–17, 1989.
- [12] H. Kautz, "The third ai summer: Aai robert s. engelmore memorial lecture," *AI Magazine*, vol. 43, no. 1, pp. 93–104, Mar. 2022. [Online]. Available: <https://ojs.aaai.org/index.php/aimagazine/article/view/19122>
- [13] D. Yu, B. Yang, D. Liu, H. Wang, and S. Pan, "A survey on neural-symbolic learning systems," *Neural Networks*, vol. 166, pp. 105–126, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0893608023003398>
- [14] J. M. Zhang, M. Harman, L. Ma, and Y. Liu, "Machine learning testing: Survey, landscapes and horizons," *IEEE Transactions on Software Engineering*, vol. 48, no. 01, pp. 1–36, 2022.
- [15] H. B. Braiek and F. Khomh, "On testing machine learning programs," *Journal of Systems and Software*, vol. 164, p. 110542, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0164121220300248>
- [16] R. C. Cardoso, G. Kourtis, L. A. Dennis, C. Dixon, M. Farrell, M. Fisher, and M. Webster, "A review of verification and validation for space autonomous systems," *Current Robotics Reports*, vol. 2, no. 3, pp. 273–283, 2021.
- [17] A. d. Garcez, S. Bader, H. Bowman, L. C. Lamb, L. de Penning, B. Illumino, H. Poon, and C. G. Zaverucha, "Neural-symbolic learning and reasoning: A survey and interpretation," *Neuro-Symbolic Artificial Intelligence: The State of the Art*, vol. 342, no. 1, p. 327, 2022.
- [18] A. Sheth, K. Roy, and M. Gaur, "Neurosymbolic artificial intelligence (why, what, and how)," *IEEE Intelligent Systems*, vol. 38, no. 3, pp. 56–62, 2023.
- [19] M. Gaur, K. Gunaratna, S. Bhatt, and A. Sheth, "Knowledge-infused learning: A sweet spot in neuro-symbolic ai," *IEEE Internet Computing*, vol. 26, no. 4, pp. 5–11, 2022.
- [20] P. Hitzler and M. K. Sarker, *Neuro-Symbolic Artificial Intelligence: The state of the art*. IOS Press, 2022.
- [21] M. K. Sarker, L. Zhou, A. Eberhart, and P. Hitzler, "Neuro-symbolic artificial intelligence," *AI Communications*, vol. 34, no. 3, pp. 197–209, 2021.
- [22] Z. Susskind, B. Arden, L. K. John, P. Stockton, and E. B. John, "Neuro-symbolic ai: An emerging class of ai workloads and their characterization," *arXiv preprint arXiv:2109.06133*, 2021.
- [23] W. Wang and Y. Yang, "Towards data-and knowledge-driven artificial intelligence: A survey on neuro-symbolic computing," *arXiv preprint arXiv:2210.15889*, 2022.
- [24] W. Gibaut, L. Pereira, F. Grassiotto, A. Osorio, E. Gadioli, A. Munoz, S. Gomes, and C. d. Santos, "Neurosymbolic ai and its taxonomy: a survey," *arXiv preprint arXiv:2305.08876*, 2023.
- [25] L. N. DeLong, R. F. Mir, M. Whyte, Z. Ji, and J. D. Fleuriot, "Neurosymbolic ai for reasoning on graph structures: A survey," *arXiv preprint arXiv:2302.07200*, 2023.
- [26] L. N. DeLong, R. F. Mir, Z. Ji, F. N. C. Smith, and J. D. Fleuriot, "Neurosymbolic ai for reasoning on biomedical knowledge graphs," *arXiv preprint arXiv:2307.08411*, 2023.
- [27] K. Hamilton, A. Nayak, B. Božić, and L. Longo, "Is neuro-symbolic ai meeting its promises in natural language processing? a structured review," *Semantic Web*, no. Preprint, pp. 1–42, 2022.
- [28] K. W. Church, "Word2vec," *Natural Language Engineering*, vol. 23, no. 1, pp. 155–162, 2017.
- [29] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.
- [30] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton *et al.*, "Mastering the game of go without human knowledge," *nature*, vol. 550, no. 7676, pp. 354–359, 2017.
- [31] 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," *CoRR*, vol. abs/1904.12584, 2019. [Online]. Available: <http://arxiv.org/abs/1904.12584>
- [32] R. Manhaeve, S. Dumancic, A. Kimmig, T. Demeester, and L. De Raedt, "Deepprolog: Neural probabilistic logic programming," *Advances in neural information processing systems*, vol. 31, 2018.
- [33] G. Lample and F. Charton, "Deep learning for symbolic mathematics," *CoRR*, vol. abs/1912.01412, 2019. [Online]. Available: <http://arxiv.org/abs/1912.01412>
- [34] L. Serafini, I. Donadello, and A. d. Garcez, "Learning and reasoning in logic tensor networks: Theory and application to semantic image interpretation," in *Proceedings of the Symposium on Applied Computing*, ser. SAC '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 125–130. [Online]. Available: <https://doi.org/10.1145/3019612.3019642>
- [35] P. Smolensky, M. Lee, X. He, W.-t. Yih, J. Gao, and L. Deng, "Basic reasoning with tensor product representations," 2016. [Online]. Available: <https://arxiv.org/abs/1601.02745>
- [36] D. Kahneman, *Thinking, fast and slow*. New York: Farrar, Straus and Giroux, 2011.
- [37] M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum, and U. Topcu, "Safe reinforcement learning via shielding," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, ser. AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018.
- [38] M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum, and U. Topcu, "Safe reinforcement learning via shielding," *arXiv preprint arXiv:1708.08611*, 2017.
- [39] I. H. Anellis, "Peirce's truth-functional analysis and the origin of the truth table," *History and Philosophy of Logic*, vol. 33, no. 1, pp. 87–97, 2012.
- [40] E. W. Beth, *Semantic Entailment and Formal Derivability*. Noord-Hollandsche, 1955.
- [41] W. Schwarz, "GitHub - wo/tpg: Tree Proof Generator — github.com," <https://github.com/wo/tpg>, [Accessed 01-Dec-2022].
- [42] E. Huaman, E. Kärle, and D. Fensel, "Knowledge graph validation," 2020. [Online]. Available: <https://arxiv.org/abs/2005.01389>
- [43] Z. H. Syed, N. Srivastava, M. Röder, and A.-C. N. Ngomo, "Copaal-an interface for explaining facts using corroborative paths," in *ISWC (Satellites)*, 2019, pp. 201–204.
- [44] J. Lehmann, D. Gerber, M. Morsey, and A.-C. Ngonga Ngomo, "Defacto-deep fact validation," in *International semantic web conference*. Springer, 2012, pp. 312–327.
- [45] A. Rula, M. Palmonari, S. Rubinacci, A.-C. Ngonga Ngomo, J. Lehmann, A. Maurino, and D. Esteves, "Tisco: Temporal scoping of facts," in *Companion Proceedings of The 2019 World Wide Web Conference*, 2019, pp. 959–960.
- [46] Z. H. Syed, M. Röder, and A.-C. Ngonga Ngomo, "Factcheck: Validating rdf triples using textual evidence," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 1599–1602.
- [47] G. Ercan, S. Elbassuoni, and K. Hose, "Retrieving textual evidence for knowledge graph facts," in *European Semantic Web Conference*. Springer, 2019, pp. 52–67.
- [48] R. Speck and A.-C. N. Ngomo, "Leopard—a baseline approach to attribute prediction and validation for knowledge graph population," *Journal of Web Semantics*, vol. 55, pp. 102–107, 2019.
- [49] A. Padia, F. Ferraro, and T. Finin, "Surface: semantically rich fact validation with explanations," *arXiv preprint arXiv:1810.13223*, 2018.
- [50] S. Metzger, S. Elbassuoni, K. Hose, and R. Schenkel, "S3k: seeking statement-supporting top-k witnesses," in *Proceedings of the 20th ACM international conference on Information and knowledge management*, 2011, pp. 37–46.
- [51] T. Rebele, F. Suchanek, J. Hoffart, J. Biega, E. Kuzey, and G. Weikum, "Yago: A multilingual knowledge base from wikipedia, wordnet, and geonames," in *The Semantic Web—ISWC 2016: 15th International Semantic Web Conference, Kobe, Japan, October 17–21, 2016, Proceedings, Part II 15*. Springer, 2016, pp. 177–185.
- [52] R. Speer, J. Chin, and C. Havasi, "Conceptnet 5.5: An open multilingual graph of general knowledge," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, no. 1, 2017.
- [53] D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. Prager *et al.*, "Building watson: An overview of the deepqa project," *AI magazine*, vol. 31, no. 3, pp. 59–79, 2010.
- [54] G. A. Miller, "Wordnet: A lexical database for english," *Commun. ACM*, vol. 38, no. 11, p. 39–41, nov 1995. [Online]. Available: <https://doi.org/10.1145/219717.219748>
- [55] N. Drenkow, N. Sani, I. Shpitser, and M. Unberath, "A systematic review of robustness in deep learning for computer vision: Mind the gap?" 2022.

- [56] M. Wicker, X. Huang, and M. Kwiatkowska, "Feature-guided black-box safety testing of deep neural networks," in *Tools and Algorithms for the Construction and Analysis of Systems: 24th International Conference, TACAS 2018, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2018, Thessaloniki, Greece, April 14-20, 2018, Proceedings, Part I* 24. Springer, 2018, pp. 408–426.
- [57] W. Ruan, X. Huang, and M. Kwiatkowska, "Reachability analysis of deep neural networks with provable guarantees," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, 7 2018, pp. 2651–2659. [Online]. Available: <https://doi.org/10.24963/ijcai.2018/368>
- [58] M. Wu, M. Wicker, W. Ruan, X. Huang, and M. Kwiatkowska, "A game-based approximate verification of deep neural networks with provable guarantees," *Theoretical Computer Science*, vol. 807, pp. 298–329, 2020, in memory of Maurice Nivat, a founding father of Theoretical Computer Science - Part II. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0304397519304426>
- [59] G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer, "Reluplex: An efficient smt solver for verifying deep neural networks," in *Computer Aided Verification: 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part I* 30. Springer, 2017, pp. 97–117.
- [60] R. Ehlers, "Formal verification of piece-wise linear feed-forward neural networks," in *Automated Technology for Verification and Analysis: 15th International Symposium, ATVA 2017, Pune, India, October 3–6, 2017, Proceedings* 15. Springer, 2017, pp. 269–286.
- [61] R. R. Bunel, I. Turkaslan, P. Torr, P. Kohli, and P. K. Mudigonda, "A unified view of piecewise linear neural network verification," *Advances in Neural Information Processing Systems*, vol. 31, 2018.
- [62] A. Lomuscio and L. Maganti, "An approach to reachability analysis for feed-forward relu neural networks," *arXiv preprint arXiv:1706.07351*, 2017.
- [63] W. Xiang, H.-D. Tran, and T. T. Johnson, "Output reachable set estimation and verification for multilayer neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 11, pp. 5777–5783, 2018.
- [64] C.-H. Cheng, G. Nührenberg, and H. Ruess, "Maximum resilience of artificial neural networks," in *Automated Technology for Verification and Analysis: 15th International Symposium, ATVA 2017, Pune, India, October 3–6, 2017, Proceedings* 15. Springer, 2017, pp. 251–268.
- [65] N. Narodytska, S. Kasiviswanathan, L. Ryzhyk, M. Sagiv, and T. Walsh, "Verifying properties of binarized deep neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.
- [66] N. Narodytska, "Formal analysis of deep binarized neural networks," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, 7 2018, pp. 5692–5696. [Online]. Available: <https://doi.org/10.24963/ijcai.2018/811>
- [67] T. Gehr, M. Mirman, D. Drachler-Cohen, P. Tsankov, S. Chaudhuri, and M. Vechev, "Ai2: Safety and robustness certification of neural networks with abstract interpretation [paper presentation]," in *IEEE Symposium on Security and Privacy (SP), San Francisco, CA, United States*. <https://doi.org/10.1109/SP.2018>, 2018.
- [68] S. Wang, K. Pei, J. Whitehouse, J. Yang, and S. Jana, "Formal security analysis of neural networks using symbolic intervals," in *27th USENIX Security Symposium (USENIX Security 18)*, 2018, pp. 1599–1614.
- [69] A. Raghunathan, J. Steinhardt, and P. Liang, "Certified defenses against adversarial examples," in *International Conference on Learning Representations*, 2018.
- [70] W. Ruan, M. Wu, Y. Sun, X. Huang, D. Kroening, and M. Kwiatkowska, "Global robustness evaluation of deep neural networks with provable guarantees for the hamming distance," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 7 2019, pp. 5944–5952. [Online]. Available: <https://doi.org/10.24963/ijcai.2019/824>
- [71] X. Huang, M. Kwiatkowska, S. Wang, and M. Wu, "Safety verification of deep neural networks," in *Computer Aided Verification: 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part I* 30. Springer, 2017, pp. 3–29.
- [72] S. Dutta, S. Jha, S. Sanakranarayanan, and A. Tiwari, "Output range analysis for deep neural networks," *arXiv preprint arXiv:1709.09130*, 2017.
- [73] L. Pulina and A. Tacchella, "An abstraction-refinement approach to verification of artificial neural networks," in *Computer Aided Verification: 22nd International Conference, CAV 2010, Edinburgh, UK, July 15-19, 2010. Proceedings* 22. Springer, 2010, pp. 243–257.
- [74] E. Wong and Z. Kolter, "Provable defenses against adversarial examples via the convex outer adversarial polytope," in *International conference on machine learning*. PMLR, 2018, pp. 5286–5295.
- [75] M. Mirman, T. Gehr, and M. Vechev, "Differentiable abstract interpretation for provably robust neural networks," in *International Conference on Machine Learning*. PMLR, 2018, pp. 3578–3586.
- [76] R. Kohavi et al., "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Ijcai*, vol. 14, no. 2. Montreal, Canada, 1995, pp. 1137–1145.
- [77] B. Efron and R. J. Tibshirani, *An introduction to the bootstrap*. CRC press, 1994.
- [78] N. Japkowicz, "Why question machine learning evaluation methods," in *AAAI workshop on evaluation methods for machine learning*, 2006, pp. 6–11.
- [79] M. Vartak, J. M. F. da Trindade, S. Madden, and M. Zaharia, "Mistiq: A system to store and query model intermediates for model diagnosis," in *Proceedings of the 2018 International Conference on Management of Data*, ser. SIGMOD '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 1285–1300. [Online]. Available: <https://doi.org/10.1145/3183713.3196934>
- [80] S. Krishnan and E. Wu, "Palm: Machine learning explanations for iterative debugging," in *Proceedings of the 2nd Workshop on Human-In-the-Loop Data Analytics*, ser. HILDA '17. New York, NY, USA: Association for Computing Machinery, 2017. [Online]. Available: <https://doi.org/10.1145/3077257.3077271>
- [81] J. Ding, X. Kang, and X.-H. Hu, "Validating a deep learning framework by metamorphic testing," in *Proceedings of the 2nd International Workshop on Metamorphic Testing*, ser. MET '17. IEEE Press, 2017, p. 28–34.
- [82] S. Nakajima, "Generalized oracle for testing machine learning computer programs," in *Software Engineering and Formal Methods: SEFM 2017 Collocated Workshops: DataMod, FAACS, MSE, CoSim-CPS, and FOCLASA, Trento, Italy, September 4-5, 2017, Revised Selected Papers* 15. Springer, 2018, pp. 174–179.
- [83] A. Dwarakanath, M. Ahuja, S. Sikand, R. M. Rao, R. J. C. Bose, N. Dubash, and S. Podder, "Identifying implementation bugs in machine learning based image classifiers using metamorphic testing," in *Proceedings of the 27th ACM SIGSOFT international symposium on software testing and analysis*, 2018, pp. 118–128.
- [84] D. Pesu, Z. Q. Zhou, J. Zhen, and D. Towey, "A monte carlo method for metamorphic testing of machine translation services," in *Proceedings of the 3rd International Workshop on Metamorphic Testing*, ser. MET '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 38–45. [Online]. Available: <https://doi.org/10.1145/3193977.3193980>
- [85] E. Breck, S. Cai, E. Nielsen, M. Salib, and D. Sculley, "The ml test score: A rubric for ml production readiness and technical debt reduction," in *2017 IEEE International Conference on Big Data (Big Data)*. IEEE, 2017, pp. 1123–1132.
- [86] S. Ma, Y. Liu, W.-C. Lee, X. Zhang, and A. Grama, "Mode: automated neural network model debugging via state differential analysis and input selection," in *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2018, pp. 175–186.
- [87] N. Hynes, D. Sculley, and M. Terry, "The data linter: Lightweight, automated sanity checking for ml data sets," in *NIPS ML Sys Workshop*, vol. 1, no. 5, 2017.
- [88] S. Schelter, D. Lange, P. Schmidt, M. Celikel, F. Biessmann, and A. Graßberger, "Automating large-scale data quality verification," *Proceedings of the VLDB Endowment*, vol. 11, no. 12, pp. 1781–1794, 2018.
- [89] K. Pei, Y. Cao, J. Yang, and S. Jana, "Deepxplore: Automated whitebox testing of deep learning systems," in *proceedings of the 26th Symposium on Operating Systems Principles*, 2017, pp. 1–18.
- [90] X. Xie, L. Ma, F. Juefei-Xu, M. Xue, H. Chen, Y. Liu, J. Zhao, B. Li, J. Yin, and S. See, "Deephunter: a coverage-guided fuzz testing framework for deep neural networks," in *Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2019, pp. 146–157.
- [91] J. Guo, Y. Zhao, H. Song, and Y. Jiang, "Coverage guided differential adversarial testing of deep learning systems," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 933–942, 2020.

- [92] D. Meng and H. Chen, “Magnet: A two-pronged defense against adversarial examples,” in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS ’17. New York, NY, USA: Association for Computing Machinery, 2017, p. 135–147. [Online]. Available: <https://doi.org/10.1145/3133956.3134057>
- [93] W. Tan, J. Renkhoff, A. Velasquez, Z. Wang, L. Li, J. Wang, S. Niu, F. Yang, Y. Liu, and H. Song, “Noisecam: Explainable ai for the boundary between noise and adversarial attacks,” *arXiv preprint arXiv:2303.06151*, 2023.
- [94] S. Qiu, Q. Liu, S. Zhou, and W. Huang, “Adversarial attack and defense technologies in natural language processing: A survey,” *Neurocomputing*, vol. 492, pp. 278–307, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0925231222003861>
- [95] N. Martins, J. M. Cruz, T. Cruz, and P. Henriques Abreu, “Adversarial machine learning applied to intrusion and malware scenarios: A systematic review,” *IEEE Access*, vol. 8, pp. 35 403–35 419, 2020.
- [96] O. Biran and C. Cotton, “Explanation and justification in machine learning: A survey,” in *IJCAI-17 workshop on explainable AI (XAI)*, vol. 8, no. 1, 2017, pp. 8–13.
- [97] C. Molnar, *Interpretable Machine Learning*, 2nd ed., 2022. [Online]. Available: <https://christophm.github.io/interpretable-ml-book>
- [98] S. Ruiter and N. D. De Graaf, “National context, religiosity, and volunteering: Results from 53 countries,” *American Sociological Review*, vol. 71, no. 2, pp. 191–210, 2006.
- [99] R. D. Cook, “Detection of influential observation in linear regression,” *Technometrics*, vol. 19, no. 1, pp. 15–18, 1977.
- [100] 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.
- [101] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, “Continuous control with deep reinforcement learning,” *arXiv preprint arXiv:1509.02971*, 2015.
- [102] J. Achiam, D. Held, A. Tamar, and P. Abbeel, “Constrained policy optimization,” in *International conference on machine learning*. PMLR, 2017, pp. 22–31.
- [103] S. Gu, L. Yang, Y. Du, G. Chen, F. Walter, J. Wang, Y. Yang, and A. Knoll, “A review of safe reinforcement learning: Methods, theory and applications,” *arXiv preprint arXiv:2205.10330*, 2022.
- [104] O. Bastani, Y. Pu, and A. Solar-Lezama, “Verifiable reinforcement learning via policy extraction,” in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018. [Online]. Available: <https://proceedings.neurips.cc/paper/2018/file/e6d8545daa42d5ced125a4bf747b3688-Paper.pdf>
- [105] Y. Tian, A. Luo, X. Sun, K. Ellis, W. T. Freeman, J. B. Tenenbaum, and J. Wu, “Learning to infer and execute 3d shape programs,” *arXiv preprint arXiv:1901.02875*, 2019.
- [106] D. Trivedi, J. Zhang, S.-H. Sun, and J. J. Lim, “Learning to synthesize programs as interpretable and generalizable policies,” *Advances in neural information processing systems*, vol. 34, pp. 25 146–25 163, 2021.
- [107] J. Huang, Z. Li, D. Jacobs, M. Naik, and S.-N. Lim, “Laser: Neuro-symbolic learning of semantic video representations,” *arXiv preprint arXiv:2304.07647*, 2023.
- [108] X. Wang, Y. Ye, and A. Gupta, “Zero-shot recognition via semantic embeddings and knowledge graphs,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [109] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka, and T. M. Mitchell, “Toward an architecture for never-ending language learning,” in *Twenty-Fourth AAAI conference on artificial intelligence*, 2010.
- [110] X. Chen, A. Shrivastava, and A. Gupta, “Neil: Extracting visual knowledge from web data,” in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 1409–1416.
- [111] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, “Imagenet large scale visual recognition challenge,” *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.
- [112] D. Zügner and S. Günnemann, “Certifiable robustness and robust training for graph convolutional networks,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 246–256.
- [113] V. P. Dwivedi, C. K. Joshi, A. T. Luu, T. Laurent, Y. Bengio, and X. Bresson, “Benchmarking graph neural networks,” *Journal of Machine Learning Research*, vol. 24, no. 43, pp. 1–48, 2023.
- [114] V. Fung, J. Zhang, E. Juarez, and B. G. Sumpter, “Benchmarking graph neural networks for materials chemistry,” *npj Computational Materials*, vol. 7, no. 1, p. 84, 2021.
- [115] N. Metropolis and S. Ulam, “The monte carlo method,” *Journal of the American statistical association*, vol. 44, no. 247, pp. 335–341, 1949.
- [116] R. Manhaeve, S. Dumancic, A. Kimmig, T. Demeester, and L. De Raedt, “Deepproblog: Neural probabilistic logic programming,” *Advances in Neural Information Processing Systems*, vol. 31, 2018.
- [117] A. D. Gordon, T. A. Henzinger, A. V. Nori, and S. K. Rajamani, “Probabilistic programming,” in *Future of Software Engineering Proceedings*, 2014, pp. 167–181.
- [118] J. Guo, Y. Jiang, Y. Zhao, Q. Chen, and J. Sun, “Dlfuzz: Differential fuzzing testing of deep learning systems,” in *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2018, pp. 739–743.
- [119] M. Bošnjak, T. Rocktäschel, J. Naradowsky, and S. Riedel, “Programming with a differentiable forth interpreter,” in *International conference on machine learning*. PMLR, 2017, pp. 547–556.
- [120] A. Odena, C. Olsson, D. Andersen, and I. Goodfellow, “Tensorfuzz: Debugging neural networks with coverage-guided fuzzing,” in *International Conference on Machine Learning*. PMLR, 2019, pp. 4901–4911.
- [121] A. Darwiche, “Sdd: A new canonical representation of propositional knowledge bases,” in *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.
- [122] E. Kern, L. M. Hilty, A. Guldner, Y. V. Maksimov, A. Filler, J. Gröger, and S. Naumann, “Sustainable software products—towards assessment criteria for resource and energy efficiency,” *Future Generation Computer Systems*, vol. 86, pp. 199–210, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0167739X17314188>
- [123] A. Guldner and J. Murach, “Measuring and assessing the resource and energy efficiency of artificial intelligence of things devices and algorithms,” in *Advances and New Trends in Environmental Informatics: Environmental Informatics and the UN Sustainable Development Goals*. Springer, 2022, pp. 185–199.
- [124] N. Ngu, N. Lee, and P. Shakarian, “Diversity measures: Domain-independent proxies for failure in language model queries,” 2023.
- [125] Z. Li, J. Huang, and M. Naik, “Scallop: A language for neurosymbolic programming,” *Proceedings of the ACM on Programming Languages*, vol. 7, no. PLDI, pp. 1463–1487, 2023.



New York. His research focuses on explainable and neurosymbolic AI.



Justus Renkhoff earned a bachelor’s and master’s degrees in Media and Computer Science from Trier University of Applied Sciences, Germany in 2019 and 2021, respectively. He is currently pursuing a doctorate degree. Previously, he worked at the Institute for Software Systems at Trier University of Applied Sciences, the Security and Optimization for Networked Globe Laboratory (SONG Lab) at Embry-Riddle Aeronautical University and University of Maryland, Baltimore County and taught as an adjunct instructor at St. Bonaventure University,

Ke Feng received master’s degree in Electrical and Computer Engineering from Embry-Riddle Aeronautical University(ERAU), Daytona Beach, Florida. She is currently pursuing a Ph.D. degree in Electrical Engineering and Computer Science at Security and Optimization for Networked Globe Laboratory, ERAU. Her major research interests include machine learning, deep learning, and the Internet of Things.



Marc Meier-Doernberg earned his master's degree in Data Science from Embry-Riddle Aeronautical University, Daytona Beach, Florida. He currently works as a Lead Analyst for United Airlines where he develops data-driven approaches to air traffic management. He previously worked for Lufthansa Group and contributed to various analytics projects. He focuses on deep learning, machine learning, and their applications in aviation.



Alvaro Velasquez is a program manager in the Innovation Information Office (I2O) of the Defense Advanced Research Projects Agency (DARPA), where he currently leads the Assured Neuro-Symbolic Learning and Reasoning (ANSR) program. Before that, Alvaro oversaw the machine intelligence portfolio of investments for the Information Directorate of the Air Force Research Laboratory (AFRL). Alvaro received his PhD in Computer Science from the University of Central Florida and is a recipient of the National Science Foundation Graduate Research

Fellowship Program (NSF GRFP) award, the University of Central Florida 30 Under 30 award, a distinguished paper award from AAAI, and best paper and patent awards from AFRL. He has co-authored 60 papers and two patents and serves as Associate Editor of the IEEE Transactions on Artificial Intelligence and his research has been funded by the Air Force Office of Scientific Research.



Houbing Herbert Song (M'12–SM'14–F'23) received the Ph.D. degree in electrical engineering from the University of Virginia, Charlottesville, VA, in August 2012.

He is currently a Professor, the Director of the NSF Center for Aviation Big Data Analytics (Planning), the Associate Director for Leadership of the DOT Transportation Cybersecurity Center for Advanced Research and Education (Tier 1 Center), and the Director of the Security and Optimization for Networked Globe Laboratory (SONG Lab,

www.SONGLab.us), University of Maryland, Baltimore County (UMBC), Baltimore, MD. Prior to joining UMBC, he was a Tenured Associate Professor of Electrical Engineering and Computer Science at Embry-Riddle Aeronautical University, Daytona Beach, FL. He serves as an Associate Editor for IEEE Transactions on Artificial Intelligence (TAI) (2023-present), IEEE Internet of Things Journal (2020-present), IEEE Transactions on Intelligent Transportation Systems (2021-present), and IEEE Journal on Miniaturization for Air and Space Systems (J-MASS) (2020-present). He was an Associate Technical Editor for IEEE Communications Magazine (2017-2020). He is the editor of ten books, the author of more than 100 articles and the inventor of 2 patents. His research interests include cyber-physical systems/internet of things, cybersecurity and privacy, and AI/machine learning/big data analytics. His research has been sponsored by federal agencies (including National Science Foundation, National Aeronautics and Space Administration, US Department of Transportation, and Federal Aviation Administration, among others) and industry. His research has been featured by popular news media outlets, including IEEE GlobalSpec's Engineering360, Association for Uncrewed Vehicle Systems International (AUVER), Security Magazine, CXOTech Magazine, Fox News, U.S. News & World Report, The Washington Times, and New Atlas.

Dr. Song is an IEEE Fellow (for contributions to big data analytics and integration of AI with Internet of Things), an Asia-Pacific Artificial Intelligence Association (AIAA) Fellow, an ACM Distinguished Member (for outstanding scientific contributions to computing), and a Full Member of Sigma Xi. Dr. Song has been a Highly Cited Researcher identified by Web of Science since 2021. He is an ACM Distinguished Speaker (2020-present), an IEEE Vehicular Technology Society (VTS) Distinguished Lecturer (2023-present) and an IEEE Systems Council Distinguished Lecturer (2023-present). Dr. Song received Research.com Rising Star of Science Award in 2022, 2021 Harry Rowe Mimno Award bestowed by IEEE Aerospace and Electronic Systems Society, and 10+ Best Paper Awards from major international conferences, including IEEE CPSCoM-2019, IEEE ICII 2019, IEEE/AIAA ICNS 2019, IEEE CBDCoM 2020, WASA 2020, AIAA/IEEE DASC 2021, IEEE GLOBECOM 2021 and IEEE INFOCOM 2022.