FuzzyKG: A Neuro-Symbolic Fuzzy Neural Network for Symbolic Knowledge Extraction

1st Paulo Vitor de Campos Souza
NOVA IMS-Information Management School
Universidade Nova de Lisboa
Lisboa, Portugal
0000-0002-7343-5844

2nd Liah Rosenfeld
NOVA IMS-Information Management School
Universidade Nova de Lisboa
Lisboa, Portugal
0009-0001-1788-0687)

***Abstract*—This paper introduces FuzzyKG, a neuro-symbolic framework for extracting interpretable symbolic knowledge from fuzzy neural networks. FuzzyKG transforms fuzzy rules learned by Gaussian membership functions and logical neurons into human-readable fuzzy axioms and organizes them into a structured Knowledge Graph (KG), explicitly representing relationships between features, linguistic terms, rule antecedents, and consequents. The framework preserves the numerical semantics of the underlying fuzzy model—such as feature relevance, rule strength, and class polarity—while exposing them in a symbolic, graph-based representation. Experimental results on benchmark datasets show that FuzzyKG maintains competitive predictive performance while producing coherent axioms and meaningful graph structures, enabling transparent analysis and supporting future neuro-symbolic extensions.**

***Index Terms*—Neuro-Symbolic, Fuzzy, Neural-Networks, Explainability, Knowledge-Graphs, Interpretability**

I. Introduction

Neuro-symbolic integration (NeSy) is an active research direction in Artificial Intelligence (AI) that aims to combine data-driven neural models with explicit symbolic structures. Neural approaches offer robustness when learning from noisy or high-dimensional data, whereas symbolic representations provide interpretability, structure, and the ability to express logical relations. This combination underlies several established frameworks, including Logic Tensor Networks (LTNs) [1], Lyrics [2], DeepProbLog [3], and NeurASP [4], which embed reasoning patterns into learning architectures. NeSy methods have been successfully applied to tasks such as semantic image interpretation [5].

As AI systems grow in complexity, the demand for models that are both accurate and interpretable has increased, particularly in domains requiring transparent or justifiable decisions [6]. At the same time, effectively integrating numerical learning and symbolic reasoning within a coherent framework remains an open challenge [7], motivating research into principled neuro-symbolic methodologies.

Fuzzy systems [8] play a key role in NeSy by encoding human-understandable linguistic concepts through membership functions and fuzzy rules, while Fuzzy Neural Networks (FNNs) combine this interpretability with neural learning capabilities. Although prior work has explored links between FNNs and logical frameworks such as LTNs [1], practical integration has remained limited, motivating a focus on extracting–rather than executing–symbolic knowledge from learned models. Accordingly, this paper introduces FuzzyKG, a neuro-symbolic framework that transforms fuzzy rules into human-readable axioms and organizes them into a Knowledge Graph (KG), explicitly capturing the relationships among features, membership functions, and outputs to make the model's internal reasoning transparent, useful and analyzable.

The method relies on Gaussian membership functions, t-norm and t-conorm operators [9], and the Moore–Penrose pseudoinverse [10] to define rule consequents, providing all parameters required for axiom extraction and KG construction. By representing feature influences, linguistic assignments, and rule polarities in a graph-based symbolic layer, FuzzyKG exposes the model's internal reasoning while preserving fuzzy semantics. This enables explicit rule dependency analysis aligned with neuro-symbolic AI objectives and, to the best of our knowledge, constitutes the first framework to extract fuzzy axioms directly from FNN-learned rules and organize them into a KG that makes the decision-making structure explicit.

The remainder of the paper is organized as follows: Section II provides the reader with the necessary background on fuzzy systems and rule extraction, while situating our contributions within the current state of the art. Section III describes the proposed FuzzyKG methodology. Section IV outlines the experimental setup, followed by the results in Section V. Finally, Section VI concludes and suggests future research directions.

II. Theoretical Background and Motivation

Fuzzy systems and FNNs [8] have long been employed to learn interpretable, rule-based models from data. These approaches construct linguistic rules that remain human-understandable by relying on Gaussian membership functions and fuzzy operators, such as t-norms and t-conorms [9]. More recent FNN variants place increased emphasis on rule extraction and model transparency [11], further strengthening their role as a bridge between numerical learning and symbolic reasoning. In the broader neuro-symbolic landscape, several frameworks integrate externally defined symbolic knowledge into neural architectures [1], [3]. However, such approaches introduce symbolic constraints during learning rather than deriving symbolic structures directly from the neural model's internal parameters.

In the general field of AI, KGs have emerged as powerful formalisms for representing entities and their semantic relations [12], [13]. Their adoption in explainability and symbolic reasoning underscores their role as effective intermediaries between learning systems and symbolic analysis pipelines. Nevertheless, despite their extensive study and demonstrated expressive power, the potential of KGs remains largely untapped. To the best of our knowledge, no existing work provides a systematic approach to (i) extract fuzzy axioms directly from the rules learned by an FNN, and (ii) organize these axioms into a KG that makes explicit the internal structure of the decision-making process. This gap constitutes the main contribution of the present work.

A. Fuzzy Neural Networks

FNNs integrate numerical learning with the interpretability of fuzzy systems by embedding membership functions and fuzzy operators into neural architectures [8]. In these models, input features are first mapped through fuzzy membership functions–typically Gaussian–yielding degrees of membership that serve as the network's intermediate representations. Hidden neurons then encode logical combinations of these membership functions, forming rule antecedents such as "x_i is $\mathrm{MF}_{i,k}$", while the output layer associates the corresponding consequents through learned weights. As a result, the model naturally induces a set of fuzzy rules whose structure can be explicitly extracted and analyzed [11].

The typical FNN architecture includes: (i) a fuzzification layer, (ii) a rule layer implementing t-norms or t-conorms, and (iii) a final layer responsible for defuzzification. This allows the network to approximate nonlinear decision boundaries while maintaining a rule-based representation accessible to symbolic inspection.

B. Axioms and Knowledge Graphs: Conceptual Definitions

a) Axioms [14]: In classical logic, an axiom is a formula assumed to be true within a theory and used to derive further consequences. Formally, for a first-order language $\mathcal{L}$ and a theory T:

$$\phi \in T \quad \text{and} \quad T \models \phi. \tag{1}$$

In fuzzy logic, axioms extend this notion by assigning graded truth degrees in $[0, 1]$ rather than binary values [15]. Such axioms rely on membership functions and fuzzy operators (t-norms, t-conorms) to express graded semantic constraints [16]. In this work, we adopt this perspective to define *fuzzy axioms* as symbolic statements derived from fuzzy rules, expressed through interval-based descriptions that retain graded semantics while remaining model-agnostic.

b) Knowledge Graphs [17]: A KG can be defined as a structured representation of entities and their semantic relations, formalized as a directed labeled graph

$$G = (V, E, \mathcal{L}), \tag{2}$$

where V is the set of nodes, $\mathcal{L}$ is the set of relation labels, and

$$E \subseteq V \times \mathcal{L} \times V \tag{3}$$

is the set of relational triples.

KGs support symbolic reasoning, explanation, and structural inspection by preserving relational dependencies in formats that are both machine-interpretable and human-readable [18]. They are widely employed to encode domain knowledge, constraints, and relational patterns, irrespective of whether such structures originate from expert knowledge or are derived from learning algorithms [17].

Together, the aforementioned definitions provide the conceptual foundation for the symbolic structures developed throughout the remainder of this paper.

III. From Fuzzy Rules to Knowledge Graphs: The FuzzyKG Framework

From a broader perspective, FuzzyKG can be viewed as a methodology that extends the classical FNN framework [19] by introducing two additional techniques within its second layer: one aimed at extracting fuzzy axioms from the learned fuzzy rules, and another devoted to constructing a KG that explicitly captures their relational structure. More specifically, as in a traditional FNN, the overall FuzzyKG pipeline can be described as a three-layer process: (a) fuzzification via Gaussian membership functions, (b) logical neurons and fuzzy rule formation, and (c) defuzzification. FuzzyKG augments layer (b) with additional steps to enable symbolic extraction and organization. For clarity, we next provide a step-by-step description of the complete FuzzyKG pipeline, including the standard components inherited from classical FNNs.

A. The FuzzyKG Pipeline

The proposed FuzzyKG model follows an eight-step pipeline structured across the three aforementioned layers. A comprehensive overview of this framework is provided in Algorithm 1, which summarizes the complete process, including training, rule extraction, axiom construction, and knowledge graph generation.

1) Layer (a): Fuzzification and Gaussian Membership Functions: The initial fuzzification stage applies a fixed set of equally distributed Gaussian membership functions to each input dimension *(Step 1)*. Then, membership activations *(Step 2)* are established in the following way: For an input variable x_j, the l-th membership function is defined as:

$$\Omega(x_j, \mu_{jl}, \sigma_{jl}) = \exp\left(-\frac{1}{2}\left(\frac{x_j - \mu_{jl}}{\sigma_{jl}}\right)^2\right). \tag{4}$$

The parameters μ_{jl} and σ_{jl} are defined *a priori* using a uniform grid partition of the input space, as commonly adopted in grid-based FNNs [8], [19]. For a predefined number of membership functions (num_mfs) per feature, the means μ_{jl} are evenly distributed over the empirical range of x_j, while the standard deviations σ_{jl} ensure sufficient overlap between adjacent Gaussians. This design guarantees complete input coverage, interpretable linguistic partitioning, and diverse rule formation, yielding a consistent and transparent fuzzification across all input dimensions.

2) Layer (b): Logical Neurons and Fuzzy Rules: Logical neurons integrate the activations of the fuzzified inputs through a general fuzzy aggregation operator, thus computing the rule activations *(Step 3)*. For a rule L with inputs $a_1, \ldots, a_n$ and feature–rule weights $w_{1L}, \ldots, w_{nL}$, the neuron output is:

$$z_L = \bigotimes_{i=1}^{n} (w_{iL}\, a_i)\,, \tag{5}$$

where $\bigotimes$ denotes a fuzzy operator that can be instantiated as:

- a t-norm ($\bigotimes = T$) to implement an AND neuron, - a t-conorm ($\bigotimes = S$) to implement an OR neuron.

This formulation unifies both logical operators in a single mathematical expression while preserving their semantic distinction through the choice of T or S. Since the rule activations have been computed for all training samples, the consequents of the fuzzy rules are then learned using the Moore–Penrose pseudo-inverse [10] *(Step 4)*.

Let $Z \in \mathbb{R}^{N \times M}$ denote the matrix of rule activations, where $Z_{nL} = z_L(x^{(n)})$ is the firing strength of rule L for sample n, M is the number of rules, and N is the number of training instances. Let $\mathbf{y} \in \mathbb{R}^N$ be the vector of target values (mapped to $[-1, 1]$ for binary problems). The consequent vector $\mathbf{v} \in \mathbb{R}^M$ is obtained by solving the linear system

$$Z\mathbf{v} \approx \mathbf{y}, \tag{6}$$

whose least–squares solution [20] is given by:

$$\mathbf{v} = Z^{+}\mathbf{y}, \tag{7}$$

where Z^{+} denotes the Moore–Penrose pseudo-inverse of Z. This closed-form solution minimizes the squared error $\|Z\mathbf{v} - \mathbf{y}\|^2$ and assigns stronger consequents to rules that consistently activate for samples of a given class. Note that for multi-class problems, this procedure is applied independently to each one-vs-all target vector.

With the rule consequents learned, the next step involves generating fuzzy rules *(Step 5)*. Let $X = \{x_1, x_2, \ldots, x_n\}$ denote the input features. For each feature x_i, the FNN selects one membership function A_i parameterized by a mean μ_i and a standard deviation σ_i, corresponding to the Gaussian in Eq. (4). A learned fuzzy rule takes the form:

$$\begin{aligned} R_L :\ & \text{IF } x_1 \text{ is } A_1 \text{ AND/OR } x_2 \text{ is } A_2\ \cdots \text{ AND/OR } x_n \text{ is } A_n \\ & \text{THEN } y = v_L, \end{aligned} \tag{8}$$

where $v_L \in [-1, 1]$ is the consequent obtained from Eq. (7). Each fuzzy set A_i associated with x_i is approximated by the interval:

$$x_i \in [\,\mu_i - \sigma_i,\ \mu_i + \sigma_i\,]. \tag{9}$$

This interval provides a human-readable representation of the region where the feature attains high membership degree.

Given the fuzzy rules extracted from the FNN, each rule is converted into an interval-based symbolic axiom by replacing the Gaussian membership conditions with their corresponding interpretable intervals *(Step 6)*. Substituting the antecedents of Rule R_L with their intervals yields the corresponding fuzzy axiom:

$$\begin{aligned} \text{Axiom}_L :\ & \text{IF } x_1 \in [\mu_{L1} - \sigma_{L1},\, \mu_{L1} + \sigma_{L1}] \\ & \text{AND/OR } \cdots \\ & \text{AND/OR } x_n \in [\mu_{Ln} - \sigma_{Ln},\, \mu_{Ln} + \sigma_{Ln}] \\ & \text{THEN } y = v_L. \end{aligned} \tag{10}$$

The resulting collection of axioms constitutes the symbolic knowledge extracted from the FNN and serves as the input to the construction of the FuzzyKG.

The symbolic knowledge extracted from the FNN comprises two complementary structures: (i) fuzzy rules expressed through membership-function conditions, and (ii) fuzzy axioms obtained by transforming these conditions into interval-based, human-readable constraints. While distinct in form, both artefacts provide semantically aligned representations of the model's reasoning process. To integrate these representations within a unified relational structure, we construct a KG *(Step 7)*–or, more specifically, a FuzzyKG–that organizes both rules and axioms as interconnected symbolic entities.

a) Rule-level representation.: Let $X = \{x_1, \ldots, x_n\}$ denote the input features, each equipped with a set of Gaussian membership functions $\mathcal{M}_i = \{\mathrm{MF}_{i1}, \ldots, \mathrm{MF}_{iK_i}\}$. A fuzzy rule learned by the FNN takes the form:

$$\text{Rule } L :\ \text{IF } x_1 \text{ is } \mathrm{MF}_{1k_1} \wedge \cdots \wedge x_n \text{ is } \mathrm{MF}_{nk_n} \text{ THEN } y = v_L, \tag{11}$$

where v_L encodes class polarity and confidence, and $w_{iL} \in [0, 1]$ quantifies the contribution of feature x_i to rule L.

Each antecedent condition is mapped to a *linguistic-term node* $T_{ik} \equiv (x_i, \mathrm{MF}_{ik})$, representing the statement that feature x_i belongs to membership region MF_{ik}. The rule antecedent is therefore represented semantically as a set of weighted links to linguistic-term nodes.

b) Axiom-level representation.: Following the procedure done in *(Step 6)*, each rule is then converted into an interval-based fuzzy axiom

$$\text{Axiom}_L :\ \text{IF } x_i \in [\mu_{Li} - \sigma_{Li},\, \mu_{Li} + \sigma_{Li}] \text{ for all } i \text{ THEN } y = v_L. \tag{12}$$

This representation abstracts away the Gaussian shape while preserving the semantic region in which each antecedent has maximal membership.

To encode this symbolic structure, we introduce *interval-term nodes*

$$I_{Li} \equiv \big(x_i,\, [\mu_{Li} - \sigma_{Li},\, \mu_{Li} + \sigma_{Li}]\big), \tag{13}$$

each corresponding to the interval-based linguistic condition of the axiom. These nodes provide a higher-level view of the antecedent semantics, bridging the gap between fuzzy sets and symbolic intervals.

c) Unified KG structure.: Both rule-level and axiom-level nodes are incorporated into a single graph

$$G = (V, E), \qquad V = V_{\text{term}} \cup V_{\text{interval}} \cup V_{\text{rule}} \cup V_{\text{axiom}} \cup V_{\text{out}}. \tag{14}$$

The node sets are:

$$V_{\text{term}} = \{T_{ik}\}, \qquad \text{(feature–MF terms)} \tag{15}$$
$$V_{\text{interval}} = \{I_{Li}\}, \qquad \text{(interval-based antecedents)} \tag{16}$$
$$V_{\text{rule}} = \{R_L\}, \qquad \text{(fuzzy rules)} \tag{17}$$
$$V_{\text{axiom}} = \{A_L\}, \qquad \text{(fuzzy axioms)} \tag{18}$$
$$V_{\text{out}} = \{O_{v_L}\}. \qquad \text{(consequents)} \tag{19}$$

Edges encode the semantic components of both structures:

$$R_L \xrightarrow{\text{HASANTECEDENT}} T_{ik}, \tag{20}$$
$$\text{weight}(R_L, T_{ik}) = w_{iL}, \tag{21}$$
$$A_L \xrightarrow{\text{HASINTERVAL}} I_{Li}, \tag{22}$$
$$R_L \xrightarrow{\text{HASCONSEQUENT}} O_{v_L}, \tag{23}$$
$$A_L \xrightarrow{\text{HASCONSEQUENT}} O_{v_L}. \tag{24}$$

Thus, the KG simultaneously captures: (i) the MF-based structure of the learned rules, (ii) the interval-based semantics of the fuzzy axioms, and (iii) the shared consequents linking the two views.

d) Class polarity and confidence.: As in the rule-based view, the consequent v_L induces

$$\text{class}(L) = \text{sign}(v_L), \qquad \text{conf}(L) = |v_L|, \tag{25}$$

which are attached to both R_L and A_L, as well as to the corresponding output node O_{v_L}.

e) Symbolic serialization.: The full graph is exportable to RDF/OWL using the predicates HASANTECEDENT, HASINTERVAL, and HASCONSEQUENT. Attributes such as *feature*, *MF index*, *interval bounds*, *impact*, *rule score*, and *confidence* are preserved as annotation properties, enabling logical querying, alignment with existing ontologies, and downstream reasoning tasks.

3) Layer (c): Defuzzification Process: Finally, Once each fuzzy rule has produced its activation level z_L and its learned consequent v_L, the model aggregates all rule contributions into a crisp output $\hat{y}$. This defuzzification step *(Step 8)* follows the standard weighted-sum formulation used in fuzzy neural networks [8], [19]:

$$\hat{y} = \sum_{L=1}^{M} f_\Gamma(z_L, v_L), \tag{26}$$

where M is the number of rules. The function f_Γ implements the rule contribution operator:

$$f_\Gamma(z_L, v_L) = z_L \cdot v_L, \tag{27}$$

i.e., the firing strength of the rule z_L scales its consequent v_L. For binary classification, $v_L \in [-1, 1]$ represents the polarity and confidence of rule L towards the two classes, while z_L quantifies the degree to which the input satisfies the antecedent. This formulation ensures that rules with higher activation and stronger consequents exert greater influence on the final decision. The defuzzified value $\hat{y}$ can then be converted into a class label via the sign function, i.e., class = sign($\hat{y}$).

Algorithm 1 FuzzyKG: Training, Axiom Extraction, and Knowledge Graph Construction

---

Input: dataset $\mathbf{X}$, target vector $\mathbf{y}$, number of membership functions num_mfs.

Layer (a): Fuzzification *Steps 1–2*

Step 1: Fuzzification. Partition each input dimension into num_mfs Gaussian membership functions. Compute uniformly spaced centers μ_{jl} and standard deviations σ_{jl}.

Step 2: Membership activations. Evaluate Gaussian membership functions (Eq. 4) for all samples, producing activations $a_{i,l}(x)$.

Layer (b): Logical Neurons and Fuzzy Rule Formation *Steps 3–7*

Step 3: Logical neuron layer. Compute rule activations using the generalized fuzzy aggregation operator (Eq. 5), yielding $z_L(x)$ for $L = 1, \ldots, M$. Or, more specifically, an activation matrix $Z \in \mathbb{R}^{N \times M}$, where $Z_{nL} = z_L(x^{(n)})$ for all samples.

Step 4: Learn rule consequents. Compute the least–squares optimal consequent vector using the Moore–Penrose pseudo-inverse:

$$\mathbf{v} = Z^{+}\mathbf{y}. \tag{28}$$

Step 5: Generate fuzzy rules. For each rule L, combine:

- the selected membership function for each feature,
- the corresponding feature impact w_{iL},
- and the learned consequent v_L,

producing rules of the form in Eq. (8).

Step 6: Convert rules into fuzzy axioms. Replace each antecedent MF_{ik} by its interval representation $[\mu_{ik} - \sigma_{ik}, \; \mu_{ik} + \sigma_{ik}]$ (Eq. 9), producing axioms of the form in Eq. (10).

Step 7: Build the FuzzyKG. Create nodes for features–MF pairs, rules, and consequents; connect them using antecedent and consequent edges with attributes $\{\texttt{impact}, \texttt{rule\_score}, \texttt{confidence}\}$ as described in Sec. III-A2.

Layer (c): Defuzzification *Step 8*

Step 8: Defuzzification process. Transform the fuzzy output to the crisp value using a Singleton (Eq. 26).

Output: Learned fuzzy rules, fuzzy axioms, the corresponding symbolic Knowledge Graph and the final output.

Overall, Algorithm 1 illustrates FuzzyKG as an extension of a classical FNN, retaining the conventional FNN pipeline (Steps 1–5 and Step 8) and augmenting it with two additional steps, namely Steps 6 and 7.

IV. Experimental Setting

This section presents the experimental objectives and evaluation setting adopted for FuzzyKG. As the core contribution of the proposed approach is symbolic knowledge extraction–comprising fuzzy rules, axioms, and KGs–the experimental analysis prioritizes interpretability and structural coherence over predictive performance. Owing to the lack of established benchmarks for assessing rule-to-axiom and axiom-to-knowledge-graph transformations, controlled classification datasets are used to study the symbolic structures produced by the model.

A. Dataset and Models

We evaluated the adaptability and interpretability of the FuzzyKG model on binary classification tasks using three datasets from the UCI Machine Learning Repository[1]—Iris [21], Mammography [22], and Liver [23]. Experiments were conducted using different neuron types (i.e., And-Neuron and Or-Neuron) and varying numbers of membership functions, and performance was assessed using interpretability-oriented metrics such as distinguishability and consistency, alongside traditional measures including accuracy, precision, F-score, and recall (further details follow).

B. Interpretability Metrics

The symbolic structures extracted by FuzzyKG (fuzzy rules, axioms, and their induced KG) are evaluated through established interpretability metrics used in fuzzy rule–based systems [24]. These metrics quantify three complementary properties: (i) how distinct the rules are (distinguishability). Below we formally define each measure.

a) Distinguishability Matrix.: The distinguishability matrix $D \in \mathbb{R}^{L\times L}$ evaluates how different two rules are in terms of their antecedent membership functions. For rule indices i and j:

$$d_{ij} = 1 - \exp\left(-\frac{(\mu_i - \mu_j)^2 + (\sigma_i - \sigma_j)^2}{2}\right), \tag{29}$$

where μ and σ denote the centers and standard deviations of the Gaussian membership functions used in the antecedents. Values close to 0 indicate highly overlapping antecedents, while values near 1 indicate well-separated rules. As usual, $d_{ii} = 1$.

b) Consistency Matrix.: The consistency matrix $C \in \mathbb{R}^{L\times L}$ quantifies the agreement among rule consequents. Given rule outputs v_i and v_j:

$$c_{ij} = \exp(-|v_i - v_j|). \tag{30}$$

Values near 1 reflect similar consequents, while values near 0 indicate divergent or contradictory conclusions. By definition, $c_{ii} = 1$. This metric is widely used in fuzzy rule–based systems to assess coherence in rule bases [25].

[1] Available at: http://archive.ics.uci.edu/ml/index.phphttp://archive.ics.uci.edu/ml/index.php

V. Results

The experimental evaluation considers two complementary dimensions of the proposed approach: (i) the predictive performance of the underlying FNN model, and (ii) the structural properties, coherence, and interpretability of the symbolic knowledge extracted in the form of fuzzy rules, axioms, and the corresponding Knowledge Graph.

Table I reports the classification results on the Iris dataset for different neuron types (AND/OR) and varying numbers of membership functions (MFs). All results are averaged over five runs.

The best performance is achieved with two membership functions, where both the AND-neuron and OR-neuron configurations attain 100% accuracy, precision, recall, and F1-score. This result indicates that the learned rule bases are sufficiently expressive to capture the class boundaries, even in the presence of Gaussian fuzzification and weighted logical aggregation.

Beyond classification, the extracted fuzzy rules and their corresponding axioms provide a transparent description of the decision process. Tables IV and V illustrate examples of the generated symbolic structures. For instance, a typical axiom derived from the rule base is:

> IF x_1 is around $[-2.11, -0.91]$ with $\sigma = 0.60$ AND x_2 is around $[-1.99, -0.82]$ with $\sigma = 0.58$, THEN the predicted class has score $v = -0.88$.

Here, the consequent v expresses both the predicted class ($\operatorname{sign}(v)$) and the confidence of the rule ($|v|$), enabling a direct symbolic interpretation of the model's internal behaviour.

Tables II and III show that the or-neuron consistently outperforms the and-neuron across both datasets, achieving its best performance with three MFs on Mammography (accuracy 0.811, precision 0.805) and two MFs on Liver (accuracy 0.721, precision 0.690). These results highlight the importance of the number of membership functions in optimizing or-neuron performance.

Focusing on the Iris dataset, the four input features—sepal length, sepal width, petal length, and petal width ($x_1, \ldots, x_4$)—are used for fuzzy rule extraction. Each feature is modeled with two Gaussian membership functions (MF1 and MF2), corresponding to lower and higher regions of the normalized feature domain, without imposing domain-specific linguistic semantics.

Both And-Neuron and Or-Neuron architectures successfully capture the separable structure of the data. As reported in Tables IV and V, the AND-based model achieves 96% accuracy, while the OR-based model reaches perfect separation (100%). This difference follows directly from the aggregation operators: AND-neurons enforce stricter conjunctive activation, whereas OR-neurons aggregate evidence more permissively, leading to broader input-space coverage. These behaviors emerge solely from the fuzzy aggregation mechanisms, independent of any prior knowledge of Iris morphology. The

NT	MFs	Acc.	F1	Rec.	Prec.
and	2	0.96 ± 0.00	0.97 ± 0.00	1.00 ± 0.00	0.94 ± 0.00
and	3	0.73 ± 0.00	0.77 ± 0.00	0.69 ± 0.00	0.87 ± 0.00
and	4	0.55 ± 0.07	0.61 ± 0.10	0.55 ± 0.13	0.69 ± 0.04
or	2	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
or	3	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
or	4	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00

TABLE I

PERFORMANCE METRICS (MEAN, STD) OF THE FNN MODEL BY NEURON TYPE (NT) AND NUMBER OF MEMBERSHIP FUNCTIONS (MFS) FOR IRIS DATASET

NT	MFs	Acc.	F1	Rec.	Prec.
and	2	0.80 ± 0.00	0.80 ± 0.00	0.82 ± 0.00	0.79 ± 0.00
and	3	0.77 ± 0.00	0.78 ± 0.00	0.81 ± 0.01	0.74 ± 0.01
and	4	0.76 ± 0.02	0.76 ± 0.02	0.78 ± 0.03	0.74 ± 0.02
or	2	0.80 ± 0.00	0.80 ± 0.00	0.85 ± 0.00	0.76 ± 0.00
or	3	0.81 ± 0.00	0.81 ± 0.00	0.81 ± 0.00	0.81 ± 0.00
or	4	0.80 ± 0.00	0.80 ± 0.00	0.80 ± 0.00	0.80 ± 0.00

TABLE II

PERFORMANCE METRICS (MEAN, STD) OF THE FNN MODEL BY NEURON TYPE (NT) AND NUMBER OF MEMBERSHIP FUNCTIONS (MFS) FOR MAMMOGRAPHY DATASET

NT	MFs	Acc.	F1	Rec.	Prec.
and	2	0.53 ± 0.00	0.36 ± 0.01	0.30 ± 0.01	0.44 ± 0.00
and	3	0.60 ± 0.07	0.57 ± 0.06	0.62 ± 0.05	0.54 ± 0.07
and	4	0.51 ± 0.07	0.49 ± 0.05	0.55 ± 0.05	0.45 ± 0.06
or	2	0.72 ± 0.00	0.67 ± 0.00	0.64 ± 0.00	0.69 ± 0.00
or	3	0.67 ± 0.00	0.62 ± 0.00	0.62 ± 0.00	0.62 ± 0.00
or	4	0.71 ± 0.00	0.66 ± 0.00	0.64 ± 0.00	0.67 ± 0.00

TABLE III

PERFORMANCE METRICS (MEAN, STD) OF THE FNN MODEL BY NEURON TYPE (NT) AND NUMBER OF MEMBERSHIP FUNCTIONS (MFS) FOR LIVER DATASET

And-Neuron with 2 MFs	
— Rules —	**Output**
IF x1 is MF1 with impact 0.64 AND x2 is MF1 with impact 0.27 AND x3 is MF1 with impact 0.04 AND x4 is MF1 with impact 0.02	[-1]
IF x1 is MF1 with impact 0.81 AND x2 is MF1 with impact 0.91 AND x3 is MF1 with impact 0.61 AND x4 is MF2 with impact 0.73	[1]
— Axioms —	
IF x1 is around [-2.74 - -1.0] with sigma 0.87 AND x2 is around [-3.54 - -1.33] with sigma 1.10 AND x3 is around [-2.17 - -0.85] with sigma 0.66 AND x4 is around [-2.08 - -0.82] with sigma 0.63	[-1]
IF x1 is around [-2.74 - -1.0] with sigma 0.87 AND x2 is around [-3.54 - -1.33] with sigma 1.10 AND x3 is around [-2.17 - -0.85] with sigma 0.66 AND x4 is around [1.08 - 2.34] with sigma 0.63	[1]

TABLE IV

GENERATED RULES AND AXIOMS FOR AND-NEURON MODEL WITH 2 MFS

resulting fuzzy axioms, derived from the learned rules, are presented as follows:

Or-Neuron with 2MFs	
— Rules —	**Output**
IF x1 is MF1 with impact 0.82 OR x2 is MF1 with impact 0.68 OR x3 is MF1 with impact 0.84 OR x4 is MF1 with impact 0.76	[-1]
IF x1 is MF1 with impact 0.98 OR x2 is MF2 with impact 0.57 OR x3 is MF2 with impact 0.98 OR x4 is MF1 with impact 0.84	[-1]
— Axioms —	
IF x1 is around [-2.74 - -1.0] with sigma 0.87 OR x2 is around [-3.54 - -1.33] with sigma 1.10 OR x3 is around [-2.17 - -0.85] with sigma 0.66 OR x4 is around [-2.08 - -0.82] with sigma 0.63	[-1]
IF x1 is around [-2.74 - -1.0] with sigma 0.87 OR x2 is around [1.99 - 4.19] with sigma 1.10 OR x3 is around [1.13 - 2.44] with sigma 0.66 OR x4 is around [-2.08 - -0.82] with sigma 0.63	[-1]

TABLE V

GENERATED RULES AND AXIOMS FOR OR-NEURON MODEL WITH 2 MFS

$$\begin{aligned}\forall x_1, x_2, x_3, x_4 \in \mathbb{R}^4 :\\ &(x_1 \in [-2.74, -1.0]) \wedge (x_2 \in [-3.54, -1.33]) \wedge\\ &(x_3 \in [-2.17, -0.85]) \wedge (x_4 \in [-2.08, -0.82])\\ &\implies y = -1\end{aligned} \tag{31}$$

$$\begin{aligned}\forall x_1, x_2, x_3, x_4 \in \mathbb{R}^4 :\\ &(x_1 \in [1.62, 3.36]) \vee (x_2 \in [1.99, 4.19]) \vee\\ &(x_3 \in [1.13, 2.44]) \vee (x_4 \in [1.08, 2.34])\\ &\implies y = 1\end{aligned} \tag{32}$$

Specifically, the extracted axioms for the Iris dataset reveal highly interpretable decision boundaries. For instance, the negative-class axiom (corresponding to Iris setosa) asserts that a sample is classified as class -1 whenever all of the following interval constraints are simultaneously satisfied:

$$\begin{aligned}
x_1 &\in [-2.74, -1.00],\\
x_2 &\in [-3.54, -1.33],\\
x_3 &\in [-2.17, -0.85],\\
x_4 &\in [-2.08, -0.82].
\end{aligned} \tag{33}$$

These intervals correspond to the membership-function centers and spreads of the "small" linguistic terms for sepal and petal dimensions, which are consistent with the compact morphology of Iris setosa.

Conversely, the positive-class axiom identifies the remaining Iris species whenever *any* of the following conditions holds:

$$\begin{aligned}
x_1 &\in [1.62, 3.36]\\
x_2 &\in [1.99, 4.19]\\
x_3 &\in [1.13, 2.44]\\
x_4 &\in [1.08, 2.34].
\end{aligned} \tag{34}$$

These ranges align with the larger sepal and petal measurements typical of Iris versicolor and Iris virginica, and they emerge directly from the structure of the Or-neuron fuzzy rule.

Knowledge Graph representation. Fuzzy rules and their extracted axioms are mapped to a Knowledge Graph (KG), where rules correspond to relational nodes, antecedent terms (feature–MF pairs) to entity nodes, and consequents to output nodes. Edges encode feature relevance and membership-function identity, while node attributes represent class polarity and confidence, making the dependencies among features, rules, and classes explicit. Complete KG visualizations are provided in the Appendix.

Interpretability metrics. Interpretability is further assessed using quantitative metrics, with full results reported in the Appendix. Distinguishability and consistency analyses (Fig. 3) indicate that most rules are structurally distinct and exhibit consistent consequents within each class, confirming the coherence of the extracted symbolic knowledge.

VI. Conclusions and Future Work

This paper presented FuzzyKG, a neuro-symbolic framework that extracts interpretable fuzzy rules from Fuzzy Neural Networks (FNNs) and organizes them into symbolic axioms and structured Knowledge Graphs (KGs). By combining fuzzy membership functions, logical neurons, and graph-based representation, FuzzyKG makes the internal reasoning of FNNs explicit, readable, and relationally structured. Experimental results show that the extracted axioms capture meaningful decision boundaries, while the resulting graphs reveal rule interactions, feature relevance, and class polarity in a transparent symbolic form. Future work will address redundancy and scalability issues through rule pruning or merging, adaptive membership-function optimization, and the generation of axioms in the original data domain. We also plan to integrate reasoning mechanisms directly over the extracted Knowledge Graphs, further strengthening the link between sub-symbolic learning and symbolic knowledge representation.

Appendix

Appendix: Knowledge Graph Visualizations

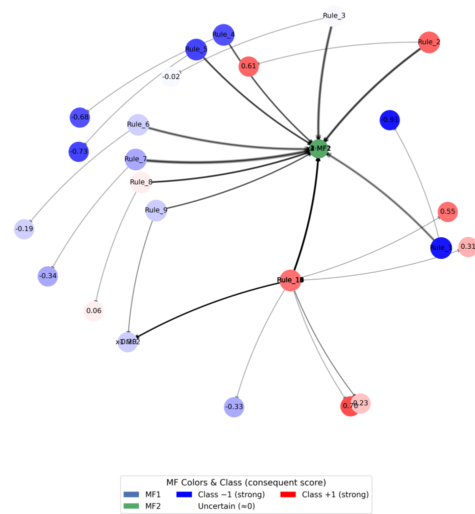

Fig. 1. Knowledge Graph constructed from the fuzzy rules extracted by the FNN for the Iris dataset. Antecedent nodes represent feature–membership-function pairs; rule nodes encode the logical neuron structure and consequent polarity; consequent nodes reflect the class score. Edge thickness denotes the impact weight associated with each antecedent.

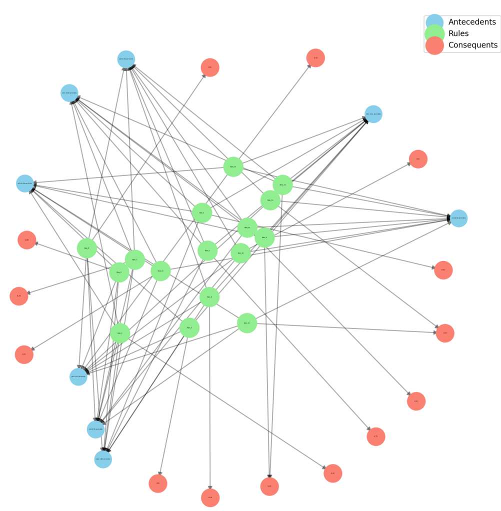

Fig. 2. Knowledge Graph representing the symbolic axioms derived from the fuzzy rules of the Iris dataset. Each interval-based antecedent is linked to the axiom node that aggregates the logical condition. Consequent nodes encode the class polarity and confidence extracted from the FNN outputs.

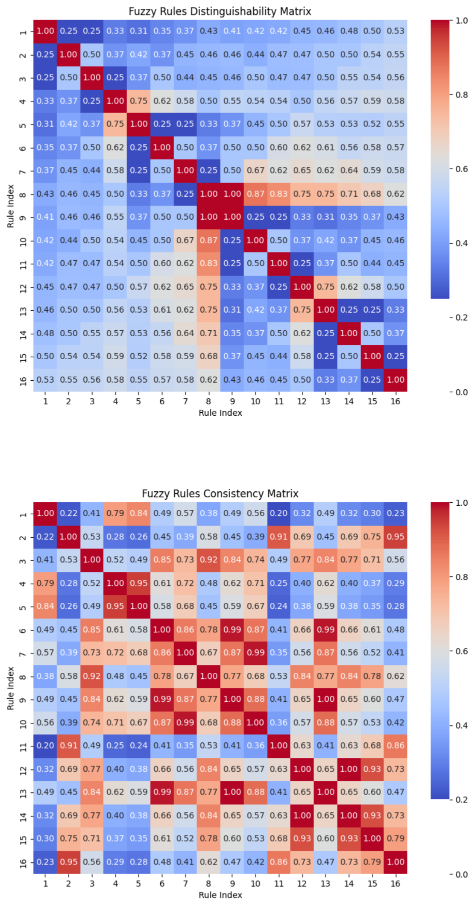

Fig. 3. Distinguishability Matrix and Consistency Matrix- Iris Dataset.

Acknowledgment

This work was supported by national funds through FCT (Fundação para a Ciência e a Tecnologia), under the project - UID/04152/2025 -(MagIC)/NOVA IMS - https://doi.org/10.54499/UID/04152/2025 (2025-01-01/2028-12-31) and UID/PRR/04152/2025 https://doi.org/10.54499/UID/PRR/04152/2025"

References

[1] S. Badreddine, A. d'Avila Garcez, L. Serafini, and M. Spranger, "Logic tensor networks," *Artificial Intelligence*, vol. 303, p. 103649, 2022.

[2] G. Marra, F. Giannini, M. Diligenti, and M. Gori, "LYRICS: A General Interface Layer to Integrate Logic Inference and deep Learning," in *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2019, Würzburg, Germany, September 16-20, 2019, Proceedings, Part II*, ser. Lecture Notes in Computer Science, U. Brefeld, É. Fromont, A. Hotho, A. J. Knobbe, M. H. Maathuis, and C. Robardet, Eds., vol. 11907, 2019, pp. 283–-298.

[3] R. Manhaeve, S. Dumancic, A. Kimmig, T. Demeester, and L. Raedt, "DeepProbLog: Neural Probabilistic Logic Programming," in *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31, 2018, pp. 3753–3763.

[4] Z. Yang, A. Ishay, and J. Lee, "NeurASP: Embracing Neural Networks into Answer Set Programming," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, C. Bessiere, Ed. ijcai.org, 2020, pp. 1755–1762.

[5] I. Donadello, L. Serafini, and A. S. d'Avila Garcez, "Logic tensor networks for semantic image interpretation," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, C. Sierra, Ed. ijcai.org, 2017, pp. 1596–1602. [Online]. Available: https://doi.org/10.24963/ijcai.2017/221

[6] R. Chimatapu, H. Hagras, A. Starkey, and G. Owusu, "Explainable ai and fuzzy logic systems," in *Theory and Practice of Natural Computing: 7th International Conference, TPNC 2018, Dublin, Ireland, December 12–14, 2018, Proceedings 7*. Springer, 2018, pp. 3–20.

[7] K. R. Thórisson, "Integrated ai systems," *Minds and Machines*, vol. 17, pp. 11–25, 2007.

[8] W. Pedrycz and F. Gomide, *Fuzzy systems engineering: toward human-centric computing*. John Wiley & Sons, 2007.

[9] M. Gupta and J. Qi, "Theory of t-norms and fuzzy inference methods," *Fuzzy Sets and Systems*, vol. 40, no. 3, pp. 431–450, 1991, fuzzy Logic and Uncertainty Modelling. [Online]. Available: https://www.sciencedirect.com/science/article/pii/016501149190171L

[10] A. Albert, *Regression and the Moore-Penrose pseudoinverse*. Elsevier, 1972.

[11] P. V. de Campos Souza, "Fuzzy neural networks and neuro-fuzzy networks: A review the main techniques and applications used in the literature," *Applied Soft Computing*, vol. 92, p. 106275, 2020.

[12] M. Gaur, K. Faldu, and A. Sheth, "Semantics of the black-box: Can knowledge graphs help make deep learning systems more interpretable and explainable?" *IEEE Internet Computing*, vol. 25, no. 1, pp. 51–59, 2021.

[13] C. Han, S. Pan, W. Que, Z. Wang, Y. Zhai, and L. Shi, "Automated localization and severity period prediction of myocardial infarction with clinical interpretability based on deep learning and knowledge graph," *Expert Systems with Applications*, vol. 209, p. 118398, 2022.

[14] I. Horrocks and S. Tobies, "Reasoning with axioms: Theory and pratice," *arXiv preprint cs/0005012*, 2000.

[15] H. Frank, "A new axiom system of fuzzy logic," *Fuzzy sets and systems*, vol. 77, no. 2, pp. 203–205, 1996.

[16] T. J. Ross, *Fuzzy logic with engineering applications*. John Wiley & Sons, 2005.

[17] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier *et al.*, "Knowledge graphs," *ACM Computing Surveys (Csur)*, vol. 54, no. 4, pp. 1–37, 2021.

[18] D. Fensel, U. Şimşek, K. Angele, E. Huaman, E. Kärle, O. Panasiuk, I. Toma, J. Umbrich, and A. Wahler, "Introduction: what is a knowledge graph?" in *Knowledge graphs: Methodology, tools and selected use cases*. Springer, 2020, pp. 1–10.

[19] P. V. de Campos Souza and L. C. B. Torres, "Regularized fuzzy neural network based on or neuron for time series forecasting," in *North American Fuzzy Information Processing Society Annual Conference*. Springer, 2018, pp. 13–23.

[20] G. H. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," in *Linear algebra*. Springer, 1971, pp. 134–151.

[21] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[22] M. Elter and A. Horsch, "Mammographic mass data," *UCI Machine Learning Repository*, 2007, https://archive.ics.uci.edu/ml/datasets/Mammographic+Mass.

[23] B. M. R. Ltd., "Liver disorders data set," *UCI Machine Learning Repository*, 1995, https://archive.ics.uci.edu/ml/datasets/Liver+Disorders.

[24] E. Lughofer, "On-line assurance of interpretability criteria in evolving fuzzy systems – achievements, new concepts and open issues," *Information Sciences*, vol. 251, pp. 22–46, 2013.

[25] ——, *Evolving fuzzy systems-methodologies, advanced concepts and applications*. Springer, 2011, vol. 53.