Preprint – IJCAI 2025: This is the accepted version made available for conference attendees.
Do not cite. The final version will appear in the IJCAI 2025 proceedings.

Human-Readable Neuro-Fuzzy Networks from Frequent Yet Discernible Patterns in Reward-Based Environments

John Wesley Hostetter, **Adittya Soukarjya Saha**, **Md Mirajul Islam**, **Tiffany Barnes** and **Min Chi**
North Carolina State University
{jwhostet, asaha4, mislam22, tmbarnes, mchi}@ncsu.edu

Abstract

We propose self-organizing and simplifying neuro-fuzzy networks (NFNs) to yield transparent human-readable policies by exploiting fuzzy information granulation and graph theory. Deriving from social network analysis, we retain only the *frequent-yet-discernible (FYD)* patterns in NFNs and apply them to reward-based scenarios. The effectiveness of NFNs from FYD patterns is shown in classic control and a real-world classroom using an intelligent tutoring system to teach students.

1 Introduction

Reinforcement Learning (RL) balances *online* exploration and exploitation to tackle reward-based environments. If exploration is prohibited, *offline* RL may train an agent to maximize an expected reward using only past data [Levine *et al.*, 2020]. Inducing policies represented by Deep Neural Networks (DNNs) with RL for complex environments has achieved great success, but suffers from a *lack of transparency* and *sample inefficiency* [Casillas *et al.*, 2013]. Fuzzy RL tackles the lack of transparency and sample inefficiency by "bootstrapping" policies with written instructions from an expert (i.e., a "warm start") via IF-THEN rules describing an approximate/imprecise causality called *fuzzy logic rules* [Lee, 1990]. Neural systems incorporating this a priori expert symbolic knowledge are a type of *neuro-symbolic network* called ***Neuro-Fuzzy Networks (NFNs)***. While the NFNs are more transparent and sample efficient, automatically building them remains an ongoing research endeavor [Aghaeipoor and Javidi, 2019]. This challenge is further exacerbated by the potential conflict between achieving *interpretability* and ensuring *accuracy* in NFNs within complex domains [Casillas *et al.*, 2013]; further compounding the difficulty in creating NFNs is the well-known *symbol grounding problem* [Harnad, 1990] found in symbolic reasoning — the issue of connecting [human] language (the symbols) to percepts [Mooney, 2008].

Our prior systematic design process (in Section 3) self-organizes NFNs using unsupervised methods and is capable of offline, model-free Fuzzy RL [Hostetter *et al.*, 2023b]. This work successfully addressed a common challenge in NFN design by preventing the fuzzy logic rules from growing linearly with the available training data. Additionally, it enabled human-in-the-loop interaction with the agent's transparent knowledge base facilitated by the NFN.

However, **two major flaws** exist in our prior work [Hostetter *et al.*, 2023b]: (1) the meaning of fuzzy logic rules' conditions cannot be adjusted during training, and (2) the number of conditions in each fuzzy logic rule grows linearly with respect to the environment's attributes. Issue (1) limits the NFN's potential for increasing the fuzzy logic rules' effectiveness, and issue (2) detrimentally affects the readability of fuzzy logic rules as the number of input attributes increases.

Our proposed work in this paper addresses issues (1) and (2) by yielding a simpler and more robust linguistic fuzzy rule base (i.e., a knowledge base) that maintains global semantics [Casillas *et al.*, 2013] without harming performance. Before fuzzy logic rule simplification, we adjust the symbols' meaning by ensuring the NFNs' internal fuzzy representations are capable of recovering the original percepts to mitigate issue (1). Then, fuzzy logic rules are simplified in a general approach compatible with Fuzzy RL by leveraging fuzzy information granulation [Zadeh, 1997] and extending social network analysis to retain only *frequent-yet-discernible (FYD)* patterns; this addresses issue (2).

To assess how effectively FYD addresses the two major issues, we compare NFNs simplified by FYD to DNNs and our prior work [Hostetter *et al.*, 2023b] in classic control tasks. Then, we empirically investigate its effectiveness in real-world higher-dimensional settings by teaching students probability principles within an intelligent tutoring system (ITS) against these same controls as well as Efficient CLAuse-wIse Rule Extraction (ECLAIRE) — an efficient, polynomial-time rule extraction algorithm to decompose large DNNs into rule-based models [Zarlenga *et al.*, 2021]. Sample code demonstrating FYD is public (MIT license) [Hostetter, 2025].[1]

2 Background

Fuzzy logic (*abbrev.* "$f.\ell.$"), $f.\ell.$ rules, linguistic variables, and their linguistic terms are all by-products of *fuzzy set theory* [Zadeh, 1965] — the mathematical study of an uncertainty called *impreciseness* [Klir and Yuan, 1995]. Often, fuzzy set theory is mistakenly compared to probability theory, but they handle different types of uncertainty — in fact, they may complement one another [Zadeh and Aliev, 2018].

[1] https://github.com/johnHostetter/IJCAI-2025-FYD

For a fuzzy set, element membership, μ, is [typically] between 0 and 1 [Klir and Yuan, 1995] — unlike "traditional" sets, where an element either belongs or it does not. Fuzzy sets more accurately reflect the nuances and spectrum to which vague or imprecise symbols/descriptions may apply to an element. For example, understanding the nuance/degree to which a person belongs to "the set of tall people" is often overlooked or ignored in set theories assuming bivalence. Here, we determine elements' membership to fuzzy sets by using the Gaussian membership function for two reasons: (1) it easily allows gradient descent [Tung *et al.*, 2011], and (2) NFNs that use Gaussians are fuzzy basis functions capable of universal function approximation as proven with the Stone-Weierstrass theorem [Wang and Mendel, 1992a].

Constrained fuzzy sets with semantics are linguistic terms, and linguistic variables can only take on values that are linguistic terms [Klir and Yuan, 1995]. An implication between the assignment of linguistic variable(s) to some linguistic term(s) (i.e., *premises*) and other linguistic variable(s) assigned to some other linguistic term(s) (i.e., *consequents*) is a $f.\ell.$ rule. Degree of applicability/activation of a $f.\ell.$ rule depends on the information's relevancy or membership in the premises. A collection of $f.\ell.$ rules is the *knowledge base*, where a $f.\ell.$ system may be built to infer outputs from input stimuli; an NFN is a computationally efficient $f.\ell.$ system capable of back-propagation [Lin and Lee, 1991]. We propose a framework to self-organize NFNs capable of policy induction by means of methods such as model-free offline RL [Kumar *et al.*, 2020] or imitation learning [Torabi *et al.*, 2018].

Assume state $\mathbf{s}$ belongs to a n-dimensional continuous space, $\prod_i^n \mathcal{S}_i$. Then, i indexes the state domain and corresponds to the i^{th} state attribute. Action space, $\mathcal{A}$, is a discrete and finite set. We will also demonstrate our methods on an example dataset (Table 1) throughout Sections 3 and 4. Building our NFNs requires no knowledge regarding the action taken or the reward received, only the state information.

ID	$\mathcal{S}_1$	$\mathcal{S}_2$	$\mathcal{S}_3$	$\mathcal{S}_4$	$\mathcal{A}$
1	0.47	1.14	0.30	1.27	a_1
2	1.40	0.02	-0.37	0.69	a_2
3	2.80	0.11	-0.09	0.42	a_1
4	-3.96	0.44	0.38	-0.28	a_1
5	4.59	1.22	0.02	1.10	a_2
6	1.34	0.56	0.22	0.53	a_2
7	-2.26	1.19	-0.24	-0.28	a_1
8	1.13	0.19	-0.10	0.48	a_2

Table 1: Example dataset; data was artificially and uniformly sampled for Cart Pole where: $\mathcal{S}_1$ is the cart's position $(-4.8, 4.8)$, $\mathcal{S}_2$ is cart's velocity $(-\inf, \inf)$, $\mathcal{S}_3$ is pole's angle (in radians) within $(\sim\!-0.418, \sim\!0.418)$, and $\mathcal{S}_4$ is pole's angular velocity $(-\inf, \inf)$. Action space, $\mathcal{A}$, is to push the cart left, a_1, or right, a_2.

3 Self-Organizing Neuro-Fuzzy Networks

Our prior work [Hostetter *et al.*, 2023b] introduced a systematic design process called CLIP-ECM-Wang-Mendel (CEW) [Hostetter and Chi, 2023] to *self-organize* an NFN from data while remaining flexible to various learning paradigms. Sections 3.1, 3.2, and 3.3 describe each step to construct components which ultimately form an NFN. Section 3.1 creates membership functions in a data-driven, incremental manner; this effectively discovers logical propositions that describe "vague" symbolic concepts (e.g., slow velocity). Section 3.2 finds "exemplary" stimuli for $f.\ell.$ rule candidacy. Section 3.3 uses results from Sections 3.1 and 3.2 to yield possible $f.\ell.$ rule premises (i.e., linguistic descriptions or qualitative assessments of the stimuli); then, a preliminary NFN is self-organized and can be trained similarly to a DNN.

3.1 Creating Membership Functions

Our $f.\ell.$ rules will map linguistic terms that describe the environment's current state to Q-values of the available actions. Fuzzy sets represent these linguistic terms (e.g., adjectives or concepts) and may be discovered using Categorical Learning Induced Partitioning (CLIP) [Tung *et al.*, 2011].

CLIP is a quick, single-pass, computationally efficient algorithm to incrementally create fuzzy sets and does not require a predetermined linguistic term count. Given the first state, $\mathbf{s}$, in the data, $\mathcal{D}$, CLIP will create a fuzzy set in each i^{th} domain as in Fig. 1.1. Initially, this fuzzy set in some i^{th} domain may be interpreted as the *concept* of being *near*, *similar*, or *approximately* s_i (the *core* of the concept). This "soft" partition allows for varying degrees of membership, such that new data is ranked according to how similar it is to s_i. The membership function, $\mu_{i,j}$, controls how we quantify similarity to the core, where j indexes the fuzzy set along the i^{th} domain. Fuzzy sets found by CLIP are defined by Gaussian membership functions, with parameters for their center (i.e., core), $c_{i,j}$, and sigma, $\sigma_{i,j}$. The value of s_i must belong to at least one fuzzy set in the state domain i with a degree exceeding ϵ. If none adequately describe s_i, then a new fuzzy set, $\mu_{i,j}$, is centered on s_i, but $\sigma_{i,j}$ depends on its neighbors, if any. If exactly one neighbor, $\mu_{i,j'}$, exists, then its $\sigma_{i,j'}$ is modified

$$\sigma_{i,j'} \leftarrow \Phi\left(\sqrt{-\frac{(c_{i,j'} - s_i)^2}{\log \kappa}}, \sigma_{i,j'}\right) \tag{1}$$

such that κ controls how sigma is adjusted and $\sigma_{i,j} = \sigma_{i,j'}$ (Fig. 1.2). Else, if $\mu_{i,j}$ has a left and a right neighbor, then apply (1) to both, and $\sigma_{i,j} = \Phi(\sigma_{i,j'}, \sigma_{i,j''})$, where j' and j'' index the left and right neighbors, respectively. The Φ is a *regulator function* such that $\Phi(\sigma_{i,j'}, \sigma_{i,j''}) = \frac{1}{2}[\sigma_{i,j'} + \sigma_{i,j''}]$ where $j' \neq j''$ and prevents malformed fuzzy sets by allowing a reasonable buffer between concepts to preserve their distinct semantic meaning. Fuzzy sets found on Table 1 using CLIP are shown in Fig. 2 with their parameters in Table 2.

$\mu_{i,j}$	$i=1$		$i=2$		$i=3$		$i=4$	
	c	σ	c	σ	c	σ	c	σ
$j=1$	0.47	3.34	1.14	0.56	0.30	0.41	1.27	0.53
$j=2$	-3.96	3.43	0.02	0.56	-0.37	0.41	0.69	0.65
$j=3$	4.59	3.34	0.56	0.56	–	–	-0.28	0.65

Table 2: The parameters of the fuzzy sets shown in Figure 2.

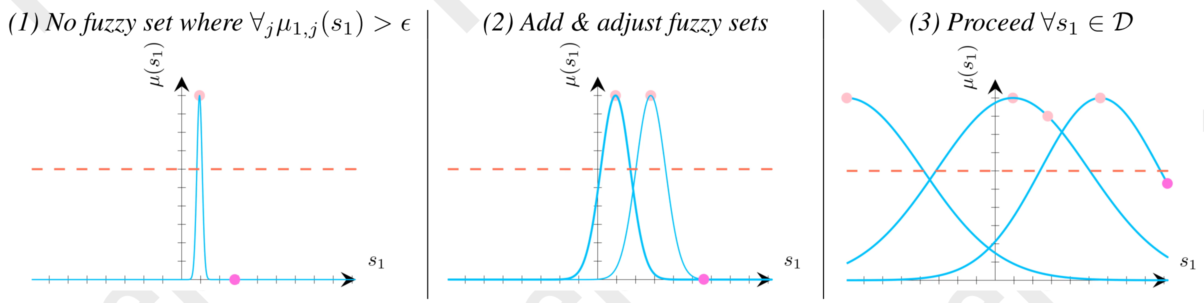

Figure 1: Applying CLIP, with $\epsilon = 0.6$ and $\kappa = 0.2$, to cart position s_1 data (pink dots) from Table 1. *(1)* $s_1 = 0.47$ creates a fuzzy set, but none exist for the next $s_1 = 1.40$; *(2)* a new fuzzy set is then made, and existing accommodate it; *(3)* continue for all remaining $s_1 \in \mathcal{D}$.

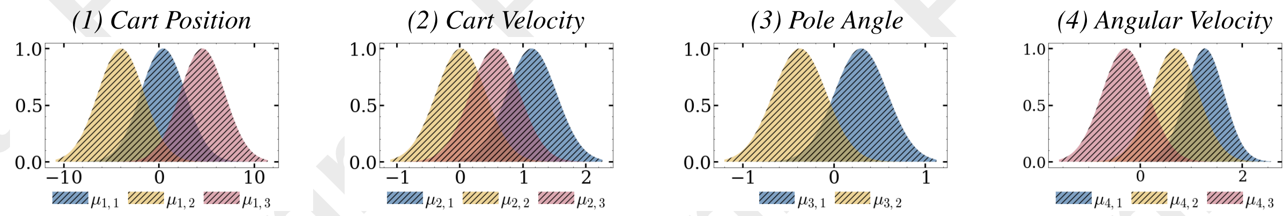

Figure 2: Fuzzy sets found (ordered by moment of creation) when applying CLIP, with $\epsilon = 0.6$ and $\kappa = 0.2$, to Table 1.

3.2 Identifying Exemplars

The task of creating $f.\ell.$ rules is closely related to identifying exemplars, clusters, or prototypes within input-output data [Angelov and Gu, 2018]. In offline RL, although states have no associated output behavior *before* learning Q-values, we can still identify exemplars or "regions of interest" for $f.\ell.$ rules by the Evolving Clustering Method (ECM) [Kasabov and Song, 2002]. ECM quickly and dynamically estimates the number of clusters within data in a single pass and finds their current centers using a *distance threshold (Dthr)*; *Dthr* affects cluster count, and subsequently, $f.\ell.$ rule count. Thus, *Dthr* can be adjusted to limit the growth of the knowledge base —a larger *Dthr* finds fewer candidates for $f.\ell.$ rules, but risks losing approximation power. Data is determined to belong to a cluster if its general Euclidean distance, defined as $||\mathbf{s} - \breve{\mathbf{s}}|| = \left(\sum_{i=1}^{n} |s_i - \breve{s}_i|^2\right)^{\frac{1}{2}} / n^{\frac{1}{2}}$ where $\mathbf{s}, \breve{\mathbf{s}} \in \mathcal{S}$, is less than the *Dthr*. Clusters' centers obtained are candidates, $\mathcal{X}$, to create $f.\ell.$ rules such that $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}$. ECM assists $f.\ell.$ rule generation by eliminating redundant states as $f.\ell.$ rules may grow linearly with respect to training data size (i.e., $|\mathcal{D}|$) [Wang and Mendel, 1992b]. Applying ECM with *Dthr* $= 0.2$ on Table 1 simply aggregates states #2 and #8 to yield $(1.265, 0.105, -0.235, 0.585)$ as there are only 8 rows.

3.3 Discovering Possible Fuzzy Logic Rules

The Wang-Mendel Method is well-established and widely used for $f.\ell.$ rule generation [Wang and Mendel, 1992b]. Given a set of training data, $\mathcal{X}$, transform each $\mathbf{x}$ to its fuzzy representation through a Cartesian product of fuzzy sets, where each fuzzy set is for a specific input dimension. To determine which fuzzy sets, for $1 \leq i \leq n$ of $\mathbf{x}$, we select the fuzzy set that x_i attains the highest degree of membership:

$$\star = \arg\max \mu_{i,j}(x_i) \text{ for } 1 \leq j \leq |\mu_i| \quad (2)$$

where $\mu_{i,j}$ is a fuzzy set and $|\mu_i|$ is the count of fuzzy sets in dimension i. A $f.\ell.$ rule links compound fuzzy sets to a decision, but we will use $\mathbf{0}$ as the decision for versatility, as this is later learned through gradient descent. Thus, given candidate $\mathbf{x}$, we make a $f.\ell.$ rule in the form: $Rule_k : (\mu_{1,\star}, \mu_{2,\star}, \ldots, \mu_{n,\star}) \Rightarrow \mathbf{0}$ where $\star$ satisfies (2) for $1 \leq i \leq n$ and $Rule_k$ means the k^{th} $f.\ell.$ rule ($k \geq 1$); rules with identical antecedents are eliminated to avoid redundancy.

From Table 1 and Fig. 2, we yield the $f.\ell.$ rules in Table 3. While these may be interpretable in our simple example, the $f.\ell.$ rules' premises grow linearly with respect to input dimensionality. If $n = 142$, then each $f.\ell.$ rule has 142 premises since CEW does not address this issue. Our proposed work in Section 4 extends CEW by fixing its shortcomings.

Rule	From ID(s)	$\mu_{1,j}$	$\mu_{2,j}$	$\mu_{3,j}$	$\mu_{4,j}$
1	1	1	1	1	1
2	2 & 8	1	2	2	2
3	3	3	2	2	2
4	4	2	3	1	3
5	5	3	1	1	1
6	6	1	3	1	2
7	7	2	1	2	3

Table 3: Each entry specifies the index j of the linguistic term associated with a particular input dimension in a $f.\ell.$ rule.

4 Proposed Methodology

Our model for maintaining ϵ-completeness (later defined) is introduced in Section 4.1, followed by FYD for simplifying $f.\ell.$ rules in Section 4.2. Finally, our NFN induces policies akin to a DNN but offers greater transparency.

4.1 Refining the Fuzzy Logic Rules

CLIP can generate high-quality fuzzy sets, but their interaction within premises may lead to subpar performance if the associated parameters are poorly chosen. In a small exploratory study, Hostetter and Chi reduced $f.\ell.$ rule count by generating $f.\ell.$ rules only for unique latent observations, using externally stored representations. This paper instead leverages the NFN's internal latent space to improve rule stability; to this end, we propose that the *fuzzy representation* of a state $\mathbf{s}$, defined as $\{\mu(\mathbf{s}) \mid \mu\}$, is of high quality if $\mathbf{s}$ can be reliably reconstructed from it.

Still, there is a significant caveat: naïvely encoding to fuzzy representations and decoding *will not work for fuzzy sets*; pure decoding is insensitive to additional constraints such as maintaining semantics. Additionally, fuzzy sets may accidentally exceed the typical domain or violate ϵ-completeness if their parameters, such as centers and widths, are allowed unrestricted modification. The ϵ-completeness property guarantees input belongs to a fuzzy set (or $f.\ell.$ rule) with a degree $\geq \epsilon$, so NFNs have ample coverage of the input space to ensure a $f.\ell.$ rule is activated strongly enough to respond accordingly. Otherwise, catastrophic numerical issues stemming from (near) zero activation of $f.\ell.$ rules within a knowledge base can cause division by zero errors.

Let K be the number of rules discovered after Section 3.3. Then, $f.\ell.$ rules are created in the form *Mamdani-Rule*$_k$: $(\mu_{1,\star}, \mu_{2,\star}, \ldots, \mu_{n,\star}) \Rightarrow (\mu_{1,\star}, \mu_{2,\star}, \ldots, \mu_{n,\star})$ where $\star$ again satisfies (2). Mamdani $f.\ell.$ rules contain fuzzy sets in their premises as well as their consequences. Naturally, they require a special form of NFN, and here, we use a Mamdani NFN with a center of sums defuzzification [Rutkowska, 2002]. We chose this approach, with Mamdani $f.\ell.$ control, as the knowledge base contained within truly embodies the intent of encoding and decoding (it's analogous, or a "$f.\ell.$ equivalent" to an approximate identity function that has a non-trivial mapping due to the non-linear internal workings of NFNs). Our remedy augments the target decoding by preserving activated rule strengths during fuzzy set tuning $(\ \mathbf{s} \mid \mu_{Rule_1}(\mathbf{s}) \ldots \mu_{Rule_K}(\mathbf{s})\ )$ where $\mu_{Rule_k}(\mathbf{s})$ is the degree of activation of $Rule_k$ when given $\mathbf{s}$. The Mamdani NFN auto-encoder's objective is to minimize the loss in reproducing $\mathbf{s}$, as well as minimize the loss between $\mu_{\textit{Mamdani-Rule}_k}(\mathbf{s})$ and $\mu_{Rule_k}(\mathbf{s})$ for all $1 \leq k \leq K$ rules. The reproduction of $\mathbf{s}$ is derived from the center of sums defuzzification of rules' consequences, which are fuzzy sets. If $\mathbf{s}$ can be reproduced from fuzzy sets in rules' consequences, then we argue that the fuzzy representation is of high quality. This promotes *cooperative and aware* fuzzy sets through their rule relations, resulting in improved reliability and stability in the self-organization process. In essence, both sets of rules cohabitate the final assembled NFN's knowledge base to help maintain consistency in the fuzzy hypercube mapping [Kosko, 1994] and avoid violating ϵ-completeness [Lee, 1990]. As observed in our experiments, adjusting individual premises is allowed if the compound premises' behavior remains consistent. If the need arises, preserving original compound rule activation strength can be omitted or updated if the system's dynamics demand it. Still, allowing compound rule activation to roam freely during Fuzzy RL tends to yield subpar performance or failure to learn. This is exacerbated if premise elimination occurs, which is introduced next in Section 4.2.

4.2 Simplifying the Knowledge Base

Fuzzy information granulation (i.e., f-granulation) theory is the magnum opus of Zadeh's legacy in soft computing [Zadeh, 1997]. This paradigm elegantly integrates fuzzy theory with numerous fields (e.g., probability, rough theory, graph theory) to leverage the full power of mathematics under a single umbrella. A compelling view in f-granulation reveals the relationship between $f.\ell.$ systems and graph theory. Although superficially simple, this has significant potential for interdisciplinary work between the fields. We leverage this viewpoint to discover $f.\ell.$ rules that have frequently occurring yet discernible premises from one another; this discernibility simplifies rules by removing redundancy, shrinking the number of premises contained in each rule, and enhancing the human readability of the knowledge while simultaneously not disrupting the rules' intended activation strengths. As such, we aptly call it *the frequent-yet-discernible (FYD) method*.

We retain premises with ***frequent*** activation to maintain ϵ-completeness. Fuzzy association analysis calculates fuzzy set frequency as the *scalar cardinality*, $\mathbf{S}$ [Chen *et al.*, 2011]. The scalar cardinality of each premise term, $\mu_{i,j}$, is computed by summing the membership degree of s_i to the fuzzy set $\mu_{i,j}$ across all states $\mathbf{s}$ in the dataset:

$$\mathbf{S} = \begin{bmatrix} \sum_{\mathbf{s}} \mu_{1,1}(s_1) & \sum_{\mathbf{s}} \mu_{1,2}(s_1) & \cdots \\ \vdots & \ddots & \\ \sum_{\mathbf{s}} \mu_{n,1}(s_n) & & \sum_{\mathbf{s}} \mu_{n,\max_i |\mu_i|}(s_n) \end{bmatrix}$$

where rows and columns correspond to variables and possible terms, respectively.

A graph is built with vertices $\mathcal{V}$ and edges $\mathcal{E}$, where $\mathcal{V} = \{\mu_{i,j} \mid (1 \leq i \leq n) \wedge (1 \leq j \leq |\mu_i|)\}$, and $\mathcal{E} = \{(\mu_{i,j}, \textit{Rule}_k) \mid \mu_{i,j} \in \textit{Rule}_k \wedge (1 \leq k \leq K) \wedge (1 \leq i \leq n) \wedge (1 \leq j \leq |\mu_i|)\}$; put simply, vertices representing the input variables' terms are connected with directed edges to rules that involve them within their premises. To find ***discernible*** (i.e., unique) premises, it is easier if we first use metrics quantifying *indiscernibility* such as each vertex's *closeness centrality*, $\mathbf{C}$ [Freeman, 1978]; $\mathbf{C}$ typically calculates a social network's structural centrality (i.e., degree of centralization in the entire network). Vertex's closeness centrality is the inverse sum of distances to all other vertices (ignoring edge direction) [Csárdi and Nepusz, 2006]. In FYD, how reachable other premises are from mutual connections (via shared $f.\ell.$ rules) determines how often two premise terms co-occur. The possibility of redundant premise terms increases as this co-occurrence becomes greater.

$\mathbf{C}$ finds how indiscernible a premise term is from others by leveraging information regarding mutual connections (i.e., indiscernibility with respect to *terms*). So we also calculate *direct usage*, $\mathbf{U}$, of each $\mu_{i,j}$ across all $f.\ell.$ rules with $\mathbf{U} = \frac{\mathbf{O}}{K}$ to further quantify premise indiscernibility (i.e., indiscernibility with respect to *rules*); the matrix $\mathbf{O}$'s entries correspond to each $\mu_{i,j}$'s out-degree (count of $f.\ell.$ rules it appears in) such that $0 \leq i \leq n$ and $0 \leq j \leq \max_i |\mu_i|$. We experimentally found that FYD has more stable global or local feature elimination in $f.\ell.$ rules by incorporating both $\mathbf{C}$ and $\mathbf{U}$.

We define *indiscernibility*, $\mathbf{IND}$, of premise term(s) with $\mathbf{C} \wedge \mathbf{U}$, where $\wedge$ is a t-norm (generalization of logical "and"). Since $\mathbf{IND} \in [0, 1]$, a premise term, $\mu_{i,j}$, is more common as $\mathbf{IND}_{i,j} \to 1$, *suggesting* it may be redundant, and vice-versa. Here, we use the Hadamard product for the t-norm where $\mathbf{IND} = \mathbf{C} \wedge \mathbf{U} = \mathbf{C} \odot \mathbf{U}$.

If *discernibility* is $\neg\mathbf{IND}$ (i.e., $1 - \mathbf{IND}$), then a higher value suggests a term is "poorly connected" or "uncommon". Assembling these components introduces the FYD formula:

$$\overbrace{\frac{\mathbf{S} - \min(\mathbf{S})}{\max(\mathbf{S}) - \min(\mathbf{S})}}^{\text{Frequent}} \overbrace{\wedge}^{\text{Yet}} \overbrace{\left(1 - \frac{\mathbf{IND} - \min(\mathbf{IND})}{\max(\mathbf{IND}) - \min(\mathbf{IND})}\right)}^{\text{Discernible}}$$

where the t-norm (i.e., "logical and" $\wedge$) may be defined in multiple manners, but for simplicity, we use the Hadamard product again. Matrix $\mathbf{S}$ and $\mathbf{IND}$ are normalized to bound them within $[0, 1]$. As $\mathbf{FYD}_{i,j} \to 1$, the premise $\mu_{i,j}$ is more frequent-yet-discernible. Example calculations of $\mathbf{C}$, $\mathbf{O}$, and $\mathbf{IND}$ are provided based on the NFN built with fuzzy sets and $f.\ell.$ rules from Fig. 2 and Table 3, respectively.

$$\overbrace{\begin{bmatrix} 0.415 & 0.362 & 0.378 \\ 0.415 & 0.334 & 0.362 \\ 0.459 & 0.415 & -- \\ 0.347 & 0.395 & 0.362 \end{bmatrix}}^{\mathbf{C}} \overbrace{\begin{bmatrix} 3 & 2 & 2 \\ 3 & 2 & 2 \\ 4 & 3 & - \\ 2 & 3 & 2 \end{bmatrix}}^{\mathbf{O}} \overbrace{\begin{bmatrix} 0.178 & 0.103 & 0.108 \\ 0.178 & 0.095 & 0.103 \\ 0.263 & 0.178 & -- \\ 0.099 & 0.169 & 0.103 \end{bmatrix}}^{\mathbf{IND}}$$

where $\mathbf{O}$'s entries are the premise term's occurrences across rules. Next, $\mathbf{S}$ and $\mathbf{FYD}$ are calculated with the example data, $\mathcal{D}$, and fuzzy sets from Table 1 and Fig. 2, respectively.

$$\overbrace{\begin{bmatrix} 5.339 & 2.283 & 3.116 \\ 3.631 & 3.891 & 4.391 \\ 4.603 & 3.829 & -- \\ 2.550 & 5.019 & 2.899 \end{bmatrix}}^{\mathbf{S}(\mathcal{D})} \overbrace{\begin{bmatrix} 0.507 & 0.000 & 0.252 \\ 0.224 & 0.526 & 0.656 \\ 0.000 & 0.257 & -- \\ 0.085 & 0.498 & 0.192 \end{bmatrix}}^{\mathbf{FYD}(\mathcal{D})}$$

Although some premise terms are considered discernible (e.g., $1 - \mathbf{IND}_{1,2} = 0.897$), when we factor in activation frequency across $\mathcal{D}$, then discernible premises (purely according to the graph) are no longer given high importance since they are infrequent (e.g., $\mathbf{FYD}_{1,2} = 0$). So, the premise term, $\mu_{1,2}$, is unnecessary, as it is seldom activated in practice despite discernibility metrics allocating high importance to it because of its rare involvement in $f.\ell.$ rules. This highlights the glaring shortcoming of naïvely simplifying rules solely on discernibility (e.g., LERS from rough set theory) as in the RSPOP family of NFNs [Ang and Quek, 2005; Das *et al.*, 2016; Iyer *et al.*, 2018] where the nature of premises' multi-valued degree of truth is totally disregarded. Similarly, relying only on activation frequency may disregard how often a premise term occurs across $f.\ell.$ rules. For example, $\mathbf{FYD}_{3,1} = 0$ because although it occurs frequently, $\mathbf{S}_{3,1} = 4.603$, it has the highest closeness centrality, $\mathbf{C}_{3,1} = 0.459$, and is used across the most rules, $\mathbf{O}_{3,1} = 4$. Our proposed $\mathbf{FYD}$ is the first work to identify this inadequacy of existing work in $f.\ell.$ rule simplification for offline Fuzzy RL, and requires no utilization of the intended outputs or targets for our $f.\ell.$ rules; this makes it appropriate for Fuzzy RL when there is no clear desired mapping yet.

Kneedle [Satopaa *et al.*, 2011] finds an appropriate knee point in the sorted values from $\mathbf{FYD}$. Then, any $\mu_{i,j}$ with a $\mathbf{FYD}_{i,j}$ below this threshold is removed, and the $f.\ell.$ rules are updated accordingly. This performs a local (and, in the extreme —global) feature selection. It may also delete $f.\ell.$ rules if all their premises are deleted, as they are then non-essential. Only the maximal frequent compound premise of $f.\ell.$ rules are kept. Kneedle calculates 0.257 as the threshold from $\mathbf{FYD}(\mathcal{D})$, and retains only 5 from 11 terms: $\mu_{1,1}$, $\mu_{2,2}$, $\mu_{2,3}$, $\mu_{3,2}$, $\mu_{4,2}$. Rule 5 is non-essential, as all its premise terms have been cut. Rules 1, 3, 4, and 7 are marked for deletion as well, as their compound premises are subsets of larger FYD premises. Specifically, rules 3 and 7 are cut since they are proper subsets of rule 2, and their activation is sufficiently captured by rule 2. Similarly, rule 4 is redundant as it is a proper subset of rule 6, so they share overlapping activation. FYD finds only rules 2 and 6 are frequent-yet-discernible from the 7 $f.\ell.$ rules in Table 3 given $\mathcal{D}$.

Rule	From ID(s)	$\mu_{1,j}$	$\mu_{2,j}$	$\mu_{3,j}$	$\mu_{4,j}$
1	1	1	–	–	–
2	**2 & 8**	**1**	**2**	**2**	**2**
3	3	–	2	2	2
4	4	–	3	–	–
5	5	–	–	–	–
6	**6**	**1**	**3**	**–**	**2**
7	7	–	–	2	–

Table 4: Only two $f.\ell.$ rules are kept (in bold).

Fuzzy Set(s)	Semantic
$\mu_{1,1}$	"slightly right of the middle"
$\mu_{2,2}$	"near zero"
$\mu_{2,3}$ & $\mu_{4,2}$	"fast (positive)"
$\mu_{3,2}$	"left leaning"

Table 5: The gap between quantitative and qualitative may be closed by attaching semantic meaning to the fuzzy sets.

A human may attach semantics to the "vague" symbols' precise definitions (Table 5) by analyzing the corresponding plotted fuzzy sets (Fig. 2). For example, rule 6 may be read as: "***if*** the cart's position ***is*** slightly right of the middle, the cart's velocity ***is*** fast (positive), and the pole angular velocity ***is*** fast (positive), ***then*** the (Q-)values of pushing the cart left or right are ___ and ___, respectively." Consequences are shown as blanks (i.e., ___) since they change (via gradient descent) based on the NFN's performance, but per Section 3.3, we initialize all consequence values to start from $\mathbf{0}$.

Fewer premises may be retained than the maximal $f.\ell.$ rule premise identified here, but finding a minimal discriminant $f.\ell.$ rule premise is left for future research. The only step remaining is to train our simplified NFN with a learning algorithm. *It is important to emphasize that an NFN, once built, is used similarly to a DNN but offers greater transparency in its decision-making.* We showcase NFN's generalizability with Kumar *et al.*'s Conservative Q-Learning (offline RL) and Behavior Cloning [Torabi *et al.*, 2018] (imitation learning).

5 Classic Control Experiments & Results

We evaluated on Cart Pole [Barto *et al.*, 1983], Mountain Car [Moore, 1990], and Two-Link Arm [Sutton, 1995].

Experiment conditions. For baselines, DNNs were trained with 7 different strategies: 1. ***Conservative Q-Learning (CQL)*** [Kumar *et al.*, 2020]; 2. ***Behavior Cloning (BC)*** [Torabi *et al.*, 2018]; 3. ***Deep Q-Learning (DQL)*** (with no offline augmentation) [Mnih *et al.*, 2015]; 4. ***Double DQL (DDQL)*** (with no offline augmentation) [Hasselt *et al.*, 2016]; 5. ***Batch Constrained Q-Learning (BCQ)*** [Fujimoto *et al.*, 2018]; 6. ***Neural Fitted Q Iteration (NFQ)*** [Riedmiller, 2005]; 7. ***Soft Actor Critic (SAC)*** [Christodoulou, 2019]. All were selected for their applicability to discrete control and offline learning potential. Additionally, we compare to NFNs self-organized using ***CLIP, ECM, and Wang-Mendel (CEW)*** [Hostetter *et al.*, 2023b; Hostetter and Chi, 2023] (as in Section 3) and trained via CQL or BC. We evaluate our proposed ***FYD method*** (Section 4) against these baselines and train its NFNs via CQL or BC. We highlight CQL and BC to showcase NFNs' ability for offline RL or imitation learning, but it is compatible with others too (e.g., SAC). DNNs had two hidden layers (256 RELU neurons each), a linear output layer, and one output neuron per possible action. All methods were optimized by *Adam* [Kingma and Ba, 2014].

Policy induction. Shared parameters were identical: α = 1.0, γ = 0.99, learning rate η = 3×10^{-4}, and the batch size was 32. For FYD or CEW, CLIP used κ = 0.2, ϵ = 0.6, and ECM's distance threshold, $Dthr$, was 0.1. Training data was collected by a DNN using DQL and experience replay while solving the given environment *online*. Each condition was run 10 times across different seeds; each algorithm was shown the same data in the same order for each seed. During each run, the amount of data available for offline training gradually increased to show how conditions behave as more data is provided. Policies were evaluated *online* for 10 episodes using OpenAI Gym and mean performance was recorded.

Results. Fig. 3.a. shows FYD remains on par with the mean performance of DNN or CEW with CQL or BC and in some cases, surpasses it (e.g., Cart Pole). Compared to the existing method of building NFNs called CEW, FYD would often obtain higher rewards and show greater stability. Since the two share the same processes for fuzzy set definition, exemplar identification, and rule generation (see Section 3), the performance difference can be attributed to our proposed changes in Section 4; the reduction of premises is shown in Fig. 3.b.

6 ITS Experiments & Results

A web-based ITS teaches ten probability principles (e.g., Bayes' Theorem). The ITS provides adaptive instructions, immediate feedback, and on-demand hints to enhance learning. Pedagogical decisions are whether a student should *solve* the next problem (***Problem-Solving (PS)***), study a *worked-out example* (***Worked-Example (WE)***), or *work collaboratively* with the ITS on the next problem (***Collaborative PS (CPS)***). An additional level of interaction between student and ITS occurs during CPS — ITS can decide to ***tell*** the student the next step or ***elicit*** the student to solve the next step.

Experiment conditions. This study was homework for an undergraduate Computer Science class. Students were told to finish in a week and that they would be graded on demonstrated effort rather than learning performance. 225 students were *randomly assigned*: FYD (N=53), CEW (N=56), DNN (N=55) and ECLAIRE (N=61). ECLAIRE is an efficient, polynomial-time rule extraction to decompose DNNs into rule-based models [Zarlenga *et al.*, 2021]. The tutor, general procedure, training materials, and questions were all the same during these studies. The training corpus provides the states, actions, and rewards for policy induction. Due to midterms and study length, 195 students finished, but 22 were excluded from the analysis due to perfect pretest scores. Final group sizes were FYD (N=39), CEW (N=38), DNN (N=44) and ECLAIRE (N=52). The students' completion rate between conditions was not significantly (sig.) different: $\chi^2(3)$ = 2.5443, p = 0.46734.

Procedure. *Textbook:* Students review probability principles. *Pretest:* Students' a priori knowledge is bench-marked with 8 single- and multiple-principle problems. *ITS training:* Students are trained with 12 problems (shown in the same order for each) with the assistance of an automated tutor (e.g., FYD, DNN). *Posttest:* Learning is evaluated with 12 problems —8 isomorphic to the pretest, with the remaining 4 as non-isomorphic multiple-principle problems. Tests were graded by 2 experienced graders in a double-blind manner.

State. 142 features that may impact student learning extracted from interaction logs are split into 5 groups: *Autonomy (10):* amount of work done by the student (e.g., steps done without help); *Temporal Situation (29):* time-related information (e.g., average time per step); *Problem-Solving (35):* current problem-solving context (e.g., problem difficulty); *Performance (57):* student's ability to solve (e.g., percentage of correct steps); *Hints (11):* student's hint usage.

Action. PS, WE, or CPS as previously described.

Reward. There is no immediate reward during tutoring, but the delayed reward is students' Normalized Learning Gain (NLG) —their learning gain irrespective of incoming competence [Abdelshiheed *et al.*, 2024; Islam *et al.*, 2024]. NLG is $\frac{posttest-pretest}{\sqrt{1-pretest}}$, where 1 is the max score for posttest and pretest score is lower than 1.

Policy induction. A hierarchy of policies was created where one policy determines what action to take on the *problem-level*, and separate policies determine whether to *elicit* or *tell* the next step during a CPS for each problem. All pedagogical policies were induced offline with CQL using 2,421 students' interaction logs over 14 semesters of classroom studies; parameters were: α = 0.1, γ = 0.99, learning rate η = 3×10^{-4}, κ = 0.2, ϵ = 0.7, with a batch size of 32. ECLAIRE policies were produced from DNNs after CQL.

Results. Effect sizes are partial eta squared (η^2) or Cohen's d. Pretests were not sig. different ($F(3, 169)$ = 0.247, p = 0.781), indicating balanced incoming competence. Regardless, students' incoming competence was factored by adjusting for pretests with a one-way ANCOVA; due to each experimental condition's strength, there was also no statistically sig. difference in students' learning ($F(3, 169)$ = 0.158, p

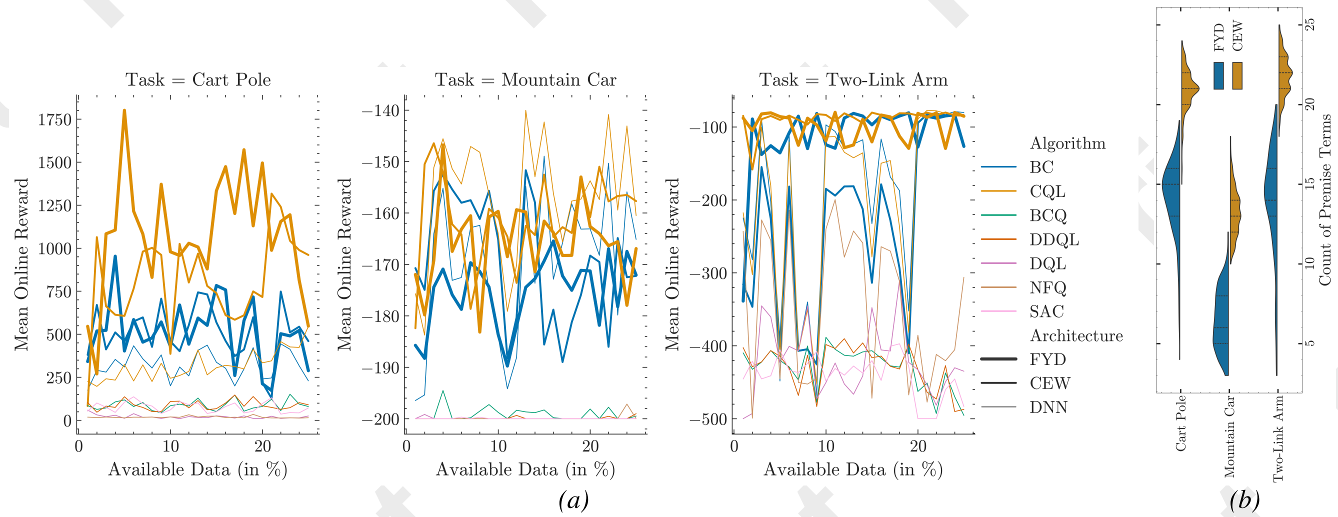

Figure 3: (a) Mean online reward of conditions' trained policies as more offline data is used; line thickness determines chosen architecture (e.g., thickest is FYD). (b) Violin plots show kernel density estimates and quartiles for the premise terms.

= 0.924), but time spent for ITS training was sig. different ($F(3, 169) = 13.047$, $p \leq 0.001$, $\eta^2 = 0.188$). Tukey-Kramer post hoc multiple comparisons found FYD had a sig. and large effect reducing the time to complete ITS training by 0.638, 0.719, 0.636 hours compared to ECLAIRE ($d = 1.374$), CEW ($d = 1.301$), and DNN ($d = 1.102$), respectively ($p \leq 0.001$ each comparison).

Model complexity. The $f.\ell.$ rule count between FYD and CEW was comparable, $t(18) = 0.316$, $p = 0.755$, $d = 0.15$, but premises' count was sig. different with pronounced effect size in favor of FYD, $t(5564) = 1720.551$, $p \leq 0.001$, $d = 46.15$. FYD had 286.8 (35.01) rules with 9.734 (3.993) premise terms, but CEW had 269.8 (92.991) rules, each with 142 premise terms. Although ECLAIRE had 2.1 (0.316) rules, their premises had 330.780 (236.032) terms. The term count in the rules' premises between FYD and ECLAIRE was sig. different with a pronounced effect (again, in favor of FYD), $t(2907) = 72.978$, $p \leq 0.001$, $d = 11.48$. For qualitative analysis, a sampled rule from each FYD and ECLAIRE in Table 6 illustrates our rules' readability. For ECLAIRE —a state-of-the-art decompositional rule extraction method for DNNs —we select the smallest rule, which still contains 26 inequality relations within its premise. Our method's ability to construct short rules conditioned upon original features offers exciting possibilities for automatic knowledge acquisition and facilitation from agent to human.

7 Related Work

Hein *et al.* self-organize NFNs for offline RL by using fuzzy particle swarm RL in a simulation to learn its parameters, but must define sought-after $f.\ell.$ rule count, and assumes it is simple to model the system's dynamics. Incorporating other logic with RL has also been researched. Relational RL (RRL) combines RL with inductive logic programming (or relational learning) to produce interpretable and generalizable policies; these may be applied to planning tasks [Džeroski *et al.*, 2001].

Condition	An Example Rule Premise
FYD *(Ours)*	IF the student only has a *FEW* hints $\wedge$ performance on this PP is *AVG* $\wedge$ correctly answer De Morgan's Law *OFTEN*
ECLAIRE	IF $[(s_1 > 0) \wedge (s_{113} > 0.5) \wedge (s_{115} > 0.857)$ $\wedge(s_6 > 0.931) \wedge (s_{86} \leq 0.056) \wedge (s_{93} \leq 0.262)]$ $\ldots \vee[(s_1 > 0) \wedge (s_{113} > 0) \wedge (s_6 \leq 0.962)$ $\wedge(s_6 > 0.956) \wedge (s_{76} > 0.925)]$

Table 6: For ECLAIRE, s_i is the i^{th} feature and values in the associated inequality are a learned threshold approximating the original DNN's decision boundary; PP refers to a probability principle.

Relational Deep RL leveraged DNNs with RRL to play StarCraft II [Zambaldi *et al.*, 2018]. Alternatives such as neural logic RL represent induced policies with first-order logic [Jiang and Luo, 2019]. Deep Symbolic RL (DSRL) derives symbolic representation from unstructured data with a DNN [Garnelo *et al.*, 2016]. Symbolic RL with Common Sense extends DSRL by generating symbolic representations prior to learning and decision-making algorithms [d'Avila Garcez *et al.*, 2018]. Other approaches aim to mimic a DNN policy by extracting an interpretable model, but encounter limitations [Bastani *et al.*, 2018; Hostetter *et al.*, 2023a].

8 Conclusion

Self-organizing NFNs are interpretable and adaptable by leveraging data for their structure and parameters. Our framework yields effective NFNs with less rule complexity but no observable loss in performance. FYD shortened the ITS training time for students and exhibited statistically higher transparency concerning premise count. Our NFNs' rules are conditioned upon the original features expressed via a linguistic medium (i.e., constrained fuzzy partitions). Their knowledge can also be transferred to/from other NFNs, opening many exciting opportunities, such as in federated learning.

Acknowledgments

This research was supported by the NSF Grants: Integrated Data-driven Technologies for Individualized Instruction in STEM Learning Environments (1726550), CAREER: Improving Adaptive Decision Making in Interactive Learning Environments (1651909), and Generalizing Data-Driven Technologies to Improve Individualized STEM Instruction by Intelligent Tutors (2013502).

References

[Abdelshiheed *et al.*, 2024] Mark Abdelshiheed, Robert Moulder, John Wesley Hostetter, Tiffany Barnes, and Min Chi. Example, nudge, or practice? Assessing metacognitive knowledge transfer of factual and procedural learners. *User Modeling and User-Adapted Interaction*, 2024.

[Aghaeipoor and Javidi, 2019] Fatemeh Aghaeipoor and Mohammad Masoud Javidi. MOKBL+MOMs: An interpretable multi-objective evolutionary fuzzy system for learning high-dimensional regression data. *Information Sciences*, 496:1–24, September 2019.

[Ang and Quek, 2005] Kai Ang and Chai Quek. RSPOP: Rough Set-Based Pseudo Outer-Product Fuzzy Rule Identification Algorithm. *Neural computation*, 17:205–43, February 2005.

[Angelov and Gu, 2018] Plamen P. Angelov and Xiaowei Gu. Empirical Fuzzy Sets. *International Journal of Intelligent Systems*, 33(2):362–395, 2018.

[Barto *et al.*, 1983] Andrew G. Barto, Richard S. Sutton, and Charles W. Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13(5):834–846, 1983.

[Bastani *et al.*, 2018] Osbert Bastani, Yewen Pu, and Armando Solar-Lezama. Verifiable reinforcement learning via policy extraction. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 2499–2509, Red Hook, NY, USA, 2018. Curran Associates Inc.

[Casillas *et al.*, 2013] Jorge Casillas, O. Cordón, Francisco Herrera Triguero, and Luis Magdalena. *Interpretability Issues in Fuzzy Modeling*. Springer, June 2013.

[Chen *et al.*, 2011] Chun-Hao Chen, Tzung-Pei Hong, and Shih-Bin Lin. Mining fuzzy temporal knowledge from quantitative transactions. In *ICSSE Proceedings*, pages 405–409, June 2011. ISSN: 2325-0925.

[Christodoulou, 2019] Petros Christodoulou. Soft actor-critic for discrete action settings. *ArXiv*, 2019.

[Csárdi and Nepusz, 2006] Gábor Csárdi and Tamás Nepusz. The igraph software package for complex network research, 2006.

[Das *et al.*, 2016] Ron Tor Das, Kai Keng Ang, and Chai Quek. ieRSPOP: A novel incremental rough set-based pseudo outer-product with ensemble learning. *Applied Soft Computing*, 46:170–186, September 2016.

[d'Avila Garcez *et al.*, 2018] Artur S. d'Avila Garcez, Aimore Dutra, and Eduardo Alonso. Towards symbolic reinforcement learning with common sense. *ArXiv*, 2018.

[Džeroski *et al.*, 2001] Sašo Džeroski, Luc De Raedt, and Kurt Driessens. Relational reinforcement learning. *Machine Learning*, 43(1):7–52, Apr 2001.

[Freeman, 1978] Linton C. Freeman. Centrality in social networks conceptual clarification. *Social Networks*, 1(3):215–239, 1978.

[Fujimoto *et al.*, 2018] Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*, 2018.

[Garnelo *et al.*, 2016] Marta Garnelo, Kai Arulkumaran, and Murray Shanahan. Towards deep symbolic reinforcement learning. *ArXiv*, 2016.

[Harnad, 1990] Stevan Harnad. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1):335–346, 1990.

[Hasselt *et al.*, 2016] Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 2094—-2100. AAAI Press, 2016.

[Hein *et al.*, 2017] Daniel Hein, Alexander Hentschel, Thomas Runkler, and Steffen Udluft. Particle swarm optimization for generating interpretable fuzzy reinforcement learning policies. *Engineering Applications of Artificial Intelligence*, 65:87–98, 10 2017.

[Hostetter and Chi, 2023] John Wesley Hostetter and Min Chi. Latent space encoding for interpretable fuzzy logic rules in continuous and noisy high-dimensional spaces. In *IEEE International Conference on Fuzzy Systems*. IEEE, 2023.

[Hostetter *et al.*, 2023a] John Wesley Hostetter, Mark Abdelshiheed, Tiffany Barnes, and Min Chi. Leveraging fuzzy logic towards more explainable reinforcement learning-induced pedagogical policies on intelligent tutoring systems. In *IEEE International Conference on Fuzzy Systems*. IEEE, 2023.

[Hostetter *et al.*, 2023b] John Wesley Hostetter, Mark Abdelshiheed, Tiffany Barnes, and Min Chi. A self-organizing neuro-fuzzy q-network: Systematic design with offline hybrid learning. In *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems*, 2023.

[Hostetter, 2025] John W. Hostetter. johnHostetter/IJCAI-2025-FYD: First release, May 2025.

[Islam *et al.*, 2024] Md Mirajul Islam, Xi Yang, John Wesley Hostetter, Adittya Soukarjya Saha, and Min Chi. A generalized apprenticeship learning framework for modeling heterogeneous student pedagogical strategies. In *Proceedings of the 17th International Conference Proceedings on Educational Data Mining*. EDM, 2024.

[Iyer *et al.*, 2018] Aparna Ramesh Iyer, Dilip K. Prasad, and Chai Hiok Quek. PIE-RSPOP: A brain-inspired pseudo-incremental ensemble rough set pseudo-outer product fuzzy neural network. *Expert Systems with Applications*, 95:172–189, April 2018.

[Jiang and Luo, 2019] Zhengyao Jiang and Shan Luo. Neural logic reinforcement learning. In *International Conference on Machine Learning*, 2019.

[Kasabov and Song, 2002] N.K. Kasabov and Qun Song. DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction. *IEEE Transactions on Fuzzy Systems*, 10(2):144–154, April 2002.

[Kingma and Ba, 2014] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

[Klir and Yuan, 1995] G. Klir and Bo Yuan. *Fuzzy sets and fuzzy logic - theory and applications*. Prentice-Hall Inc., Upper Saddle River, New Jersey, 1995.

[Kosko, 1994] B. Kosko. Fuzzy systems as universal approximators. *IEEE Transactions on Computers*, 43(11):1329–1333, 1994.

[Kumar *et al.*, 2020] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2020. Curran Associates Inc.

[Lee, 1990] C.C. Lee. Fuzzy logic in control systems: fuzzy logic controller. I & II. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(2):404–435, 1990.

[Levine *et al.*, 2020] Sergey Levine, Aviral Kumar, G. Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *ArXiv*, 2020.

[Lin and Lee, 1991] C.-T. Lin and C.S.G. Lee. Neural-network-based fuzzy logic control and decision system. *IEEE Transactions on Computers*, 40(12):1320–1336, December 1991.

[Mnih *et al.*, 2015] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, and et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[Mooney, 2008] Raymond J. Mooney. Learning to connect language and perception. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 3*, pages 1598—1601. AAAI Press, 2008.

[Moore, 1990] Andrew Moore. Efficient memory-based learning for robot control. Technical report, Carnegie Mellon University, Pittsburgh, PA, 1990.

[Riedmiller, 2005] Martin Riedmiller. Neural fitted q iteration – first experiences with a data efficient neural reinforcement learning method. In João Gama, Rui Camacho, Pavel B. Brazdil, Alípio Mário Jorge, and Luís Torgo, editors, *Machine Learning: ECML 2005*, pages 317–328, Berlin, Heidelberg, 2005. Springer.

[Rutkowska, 2002] Danuta Rutkowska. *Neuro-Fuzzy Architectures Based on the Mamdani Approach*, pages 105–126. Physica-Verlag HD, Heidelberg, 2002.

[Satopaa *et al.*, 2011] Ville Satopaa, Jeannie Albrecht, David Irwin, and Barath Raghavan. Finding a ”kneedle” in a haystack: Detecting knee points in system behavior. In *31st International Conference on Distributed Computing Systems Workshops*, pages 166–171, 2011.

[Sutton, 1995] Richard S Sutton. Generalization in reinforcement learning: Successful examples using sparse coarse coding. In *Advances in Neural Information Processing Systems*, volume 8. MIT Press, 1995.

[Torabi *et al.*, 2018] Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 4950—4957. AAAI Press, 2018.

[Tung *et al.*, 2011] Sau Wai Tung, Chai Quek, and Cuntai Guan. SaFIN: A Self-Adaptive Fuzzy Inference Network. *IEEE Transactions on Neural Networks*, 22(12):1928–1940, December 2011.

[Wang and Mendel, 1992a] L.-X. Wang and J.M. Mendel. Fuzzy basis functions, universal approximation, and orthogonal least-squares learning. *IEEE Transactions on Neural Networks*, 3(5):807–814, 1992.

[Wang and Mendel, 1992b] L.-X. Wang and J.M. Mendel. Generating fuzzy rules by learning from examples. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(6):1414–1427, November 1992.

[Zadeh and Aliev, 2018] Lotfi Zadeh and Rafik Aliev. *Fuzzy Logic Theory and Applications: Part I and II*. World Scientific, New Jersey, 2018.

[Zadeh, 1965] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.

[Zadeh, 1997] Lotfi A. Zadeh. Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy Sets and Systems*, 90(2):111–127, 1997.

[Zambaldi *et al.*, 2018] Vinícius Flores Zambaldi, David Raposo, Adam Santoro, Victor Bapst, Yujia Li, Igor Babuschkin, Karl Tuyls, David P. Reichert, Timothy P. Lillicrap, Edward Lockhart, Murray Shanahan, Victoria Langston, Razvan Pascanu, Matthew M. Botvinick, Oriol Vinyals, and Peter W. Battaglia. Relational deep reinforcement learning. *ArXiv*, 2018.

[Zarlenga *et al.*, 2021] Mateo Espinosa Zarlenga, Zohreh Shams, and Mateja Jamnik. Efficient decompositional rule extraction for deep neural networks. *ArXiv*, 2021.