BRIDGING THE GAP BETWEEN FUZZY LOGIC AND LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Connectionism and symbolism are two influential traditions in artificial intelligence. Large language models (LLMs) exemplify connectionist learning, while fuzzy logic expresses numerical reasoning through graded linguistic rules. We introduce Fuzzy-LLM, a framework that connects these traditions by using a pretrained LLM to construct an interpretable fuzzy classifier from labeled numerical examples without parameter fine-tuning. After feature normalization, the LLM describes values using five predefined linguistic terms and generates IF–THEN rules. For rules with identical antecedents but different class consequents, hard correction selects one class, whereas soft correction retains candidate classes with graded certainty. The rules are translated into a Mamdani inference system that makes predictions without further LLM calls. For each prediction, membership degrees, activated rules, and class scores trace the decision from numerical input to output. Across Iris, Ecoli, Glass, Seeds, and UKM, the best observed accuracies among the evaluated configurations are 94.67%, 82.14%, 61.68%, 94.29%, and 76.73%, respectively. Fuzzy-LLM exceeds the recorded direct-LLM baselines on all five datasets. This combination of language-guided rule construction and explicit fuzzy inference supports inspectable tabular classification.

1 INTRODUCTION

Two influential traditions in artificial intelligence are *connectionism*, which learns patterns through networks, and *symbolism*, which represents knowledge in explicit structures and rules. Large language models (LLMs) exemplify the former: pretraining gives them broad linguistic and domain knowledge (Zhao et al., 2023). Fuzzy logic offers a form of the latter in which numerical values have graded membership in linguistic concepts and decisions can be expressed as IF–THEN rules (Zadeh, 1965; Mamdani & Assilian, 1975). Connecting these traditions is appealing for tabular classification: an LLM can help describe patterns in labeled data, and a fuzzy system can turn the resulting descriptions into an inspectable decision procedure.

The connection is useful because numerical decisions demand more than a plausible verbal description. A classifier for Iris, for example, must combine sepal and petal measurements to distinguish species, including observations near class boundaries. LLMs can draw on qualitative descriptions of the species, but their tabular predictions depend on how rows and feature values are presented (Sui et al., 2024). Direct prompting therefore does not by itself yield a reusable rule base showing how each feature contributes to a decision. We ask whether a pretrained LLM can instead construct a classifier whose predictions are carried out by explicit numerical rules.

Fuzzy sets provide the interface between numerical observations and linguistic descriptions. A measured feature can belong to terms such as *small*, *medium*, and *large* to different degrees; an IF–THEN rule combines these terms and associates them with a class. This makes it possible to inspect both the rule itself and how strongly an observation activates it. Unlike a hard assignment to one label, graded membership also preserves information about values near linguistic boundaries. Prior work has combined fuzzy representations with learned models (Lu et al., 2024); here the LLM helps construct the rule base, while the fuzzy system performs inference from the original numerical input.

We introduce **Fuzzy-LLM**, which separates offline rule construction from online prediction. Given labeled numerical examples, we normalize each feature and prompt a frozen LLM to express val-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

ues using predefined linguistic terms and generate IF–THEN rules. We group rules with identical antecedents and compare three ways to handle different class consequents: retain the original rules, use hard correction to select one class, or use soft correction to retain candidate classes with graded certainty. The resulting rules are translated into an executable Mamdani system. At prediction time, this system evaluates memberships, activates rules, aggregates class outputs, and makes a decision without further LLM calls. Its rule base and prediction trace can be inspected for individual observations.

Related approaches use language models differently. TabLLM serializes rows for language-model prediction (Hegselmann et al., 2023), whereas FeatLLM generates rule-based features for a downstream predictor (Han et al., 2024). TabPFN provides in-context tabular classification with a pretrained model (Hollmann et al., 2023). Fuzzy-LLM focuses on a standalone fuzzy classifier whose conditions, consequents, and activated rules are readable. We evaluate it on Iris, Ecoli, Glass, Seeds, and UKM against direct LLM prediction, conventional classifiers, and fine-tuned Qwen models. The best observed configuration on each dataset exceeds the recorded direct-LLM baseline; conventional classifiers are stronger on Ecoli, Glass, and UKM. Correction improves Iris accuracy but reduces Glass accuracy, showing why both corrected and original rules must be evaluated. The dataset-wise maxima are exploratory because configurations were identified from reported test accuracies.

Our contributions are an offline procedure for turning LLM-generated linguistic descriptions into an executable fuzzy classifier, two strategies for handling conflicting consequents, and a five-dataset study of accuracy, component choices, and construction cost. The resulting predictions can be traced through membership degrees, activated rules, and class scores.

2 Related Work

Tabular prediction with language models. Direct prompting asks an LLM to infer a class from a serialized row, as in TabLLM (Hegselmann et al., 2023); its results depend on the representation of the table and the prompt (Sui et al., 2024). FeatLLM instead uses an LLM to generate rule-based features for a separate predictor (Han et al., 2024). Fuzzy-LLM is closer to this offline-construction setting: the LLM helps build a classifier, and the deployed model does not query it. Specialized tabular predictors such as TabPFN and TabM provide important accuracy reference points, although their predictions are not expressed as linguistic rules (Hollmann et al., 2023; Gorishniy et al., 2025).

Fuzzy and symbolic inference. Fuzzy sets map numerical values to graded linguistic terms (Zadeh, 1965), and Mamdani systems combine those terms through inspectable rules (Mamdani & Assilian, 1975). ANFIS learns fuzzy input–output mappings from data (Jang, 1993). Surveys cover extensions that combine fuzzy and deep learning (Lu et al., 2024; Zheng et al., 2022). Wang et al. study fuzzy neural networks for image clustering (Wang et al., 2021; 2023), image classification (Wang et al., 2024), and feature fusion (Wang et al., 2025); our task is numerical tabular classification. Neural Logic Machines implement relational inference (Dong et al., 2019). Other systems translate language into executable structures or delegate reasoning to symbolic solvers (Cheng et al., 2023; Pan et al., 2023); our executable structure is a fuzzy classifier.

LLMs and fuzzy reasoning. Prior work has combined LLMs with fuzzy components for fact checking and activity understanding (Bangerter et al., 2025; Wu et al., 2023). Yet LLMs can struggle with fuzzy quantifiers (Li et al., 2024). Self-Refine uses iterative model feedback (Madaan et al., 2023), while intrinsic correction without external feedback can fail (Huang et al., 2024). These findings motivate our comparison of uncorrected rules with hard and soft conflict correction. A visible conflict signals ambiguity in a linguistic rule base; it does not prove which consequent is correct.

3 Proposed Method

Figure 1 separates Fuzzy-LLM into offline construction and online prediction. A frozen LLM generates linguistic rules from labeled examples, handles conflicting consequents, and translates the rules into executable fuzzy control statements (Stages 1–3). A Mamdani system then classifies new observations without LLM calls (Stage 4). The rule base and prediction steps remain available for inspection.

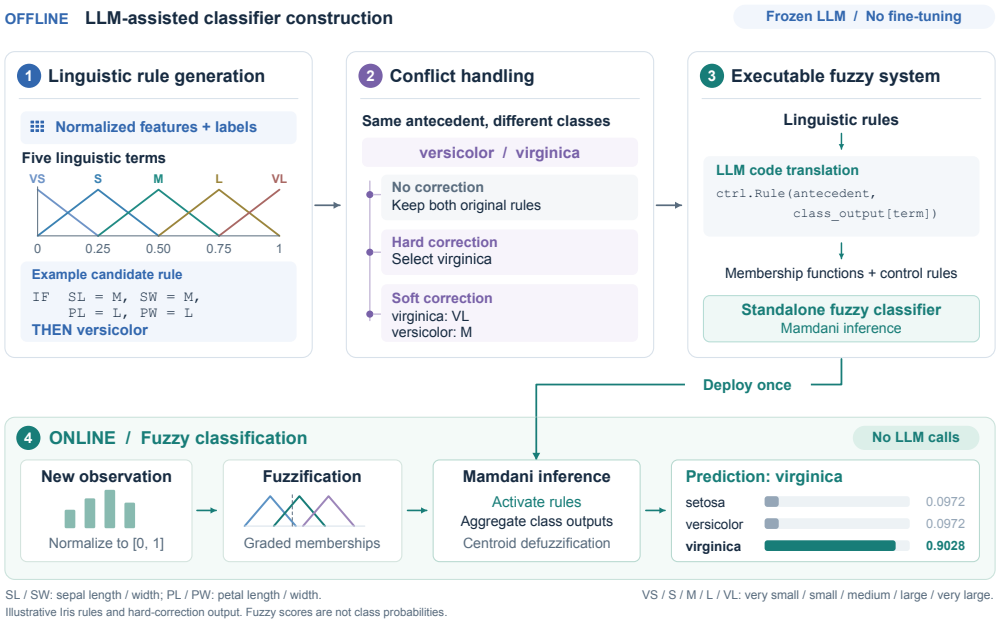


Figure 1: Overview of Fuzzy-LLM. Stages 1–3 construct an executable fuzzy classifier offline using a frozen LLM. Stage 4 performs online prediction through Mamdani fuzzy inference without additional LLM calls.

3.1 Problem Formulation

Let $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ be a labeled training set, where $\mathbf{x}_i \in \mathbb{R}^D$ contains D numerical features and $y_i \in \mathcal{Y} = \{1, \dots, C\}$ is a class label. Feature-wise normalization maps each observation to $\widetilde{\mathbf{x}}_i = T(\mathbf{x}_i) \in [0,1]^D$. Feature names and class descriptions provide the semantic context for the pretrained LLM $\mathcal{M}_\theta$, whose parameters θ remain fixed throughout the procedure. Our objective is to construct an explicit fuzzy classifier $f_{\mathcal{R}}$ from $\mathcal{D}$, with a rule base $\mathcal{R}$ that can be inspected independently of the LLM.

3.2 Linguistic Representation and Rule Generation

We use five ordered linguistic terms, $\mathcal{L} = \{\ell_0, \dots, \ell_4\}$, corresponding to *very small*, *small*, *medium*, *large*, and *very large*. Their reference centers are

$$(m_0, m_1, m_2, m_3, m_4) = (0, 0.25, 0.5, 0.75, 1). \tag{1}$$

During rule generation, a normalized feature value v is assigned the term associated with its nearest center:

$$q(v) = \ell_{j^\star}, \qquad j^\star \in \operatorname*{arg\,min}_{j \in \{0,\dots,4\}} |v - m_j|. \tag{2}$$

For example, $q(0.22) =$ *small* because 0.25 is the nearest reference center. These descriptions express magnitude relative to the normalized feature scale.

The generation prompt supplies the linguistic terms, their centers, and a worked example. It instructs the LLM to calculate the absolute distances, select the nearest term for each feature, and combine the resulting antecedent with the observed class label. The intended rule for training observation i is

$$\begin{aligned} R_i: \quad & \text{IF } \widetilde{x}_1 \text{ is } a_{i1} \text{ AND } \cdots \text{ AND } \widetilde{x}_D \text{ is } a_{iD}, \\ & \text{THEN the class is } y_i, \qquad a_{id} = q(\widetilde{x}_{id}). \end{aligned} \tag{3}$$

Thus, the class consequent is supplied by the labeled example, while the LLM expresses the numerical antecedent in the prescribed linguistic form. Collecting the distinct generated rules produces the initial rule base $\mathcal{R}_0$.

The assignment in Equation (2) specifies the prompt's conversion procedure; it does not guarantee that an LLM will execute every numerical comparison correctly. It also differs from inference-time fuzzification. During generation, a single linguistic label describes each feature. During prediction, the original normalized numerical value is evaluated against membership functions and can activate multiple linguistic terms to different degrees. We consider triangular, trapezoidal, and Gaussian membership functions within this common representation.

3.3 Correction of Conflicting Consequents

Let $\mathbf{a} = (a_1, \ldots, a_D)$ denote a linguistic antecedent. We group initial rules by their antecedents and collect the associated classes:

$$\mathcal{C}(\mathbf{a}) = \{y_i : (\mathbf{a}_i, y_i) \in \mathcal{R}_0,\ \mathbf{a}_i = \mathbf{a}\}. \tag{4}$$

A group is conflicting when $|\mathcal{C}(\mathbf{a})| > 1$. Such conflicts can arise because different numerical observations share the same linguistic representation. For example, both 0.38 and 0.62 are assigned to *medium*, although the underlying observations may belong to different classes. Errors in the LLM's numerical-to-linguistic conversion can also contribute to conflicting rules.

For each conflicting group, the correction prompt presents the antecedent, candidate classes, and relevant feature and class descriptions. It asks the LLM to assess the candidate consequents using its pretrained knowledge. We consider the following three settings.

No correction. The original antecedent–class rules are retained, including rules that share an antecedent but predict different classes. This setting provides a reference for assessing whether LLM-guided revisions improve the classifier.

Hard correction. The LLM is instructed to select one class from the conflicting candidates:

$$\begin{aligned} \widehat{c}_{\mathbf{a}} &= \mathcal{M}_\theta^{\mathrm{hard}}\big(\mathbf{a}, \mathcal{C}(\mathbf{a}); \mathcal{P}_{\mathrm{hard}}\big), \\ \widehat{c}_{\mathbf{a}} &\in \mathcal{C}(\mathbf{a}), \end{aligned} \tag{5}$$

where $\mathcal{P}_{\mathrm{hard}}$ denotes the correction prompt. The conflicting rules are replaced by a rule with the same antecedent and the selected consequent. Antecedents associated with a single class remain unchanged.

Soft correction. The LLM retains the candidate classes and assigns each a linguistic certainty level:

$$\begin{aligned} &\{(c, \lambda_{\mathbf{a},c}) : c \in \mathcal{C}(\mathbf{a})\} \\ &\quad = \mathcal{M}_\theta^{\mathrm{soft}}\big(\mathbf{a}, \mathcal{C}(\mathbf{a}); \mathcal{P}_{\mathrm{soft}}\big), \end{aligned} \tag{6}$$

where $\lambda_{\mathbf{a},c} \in \mathcal{L}$. These levels specify fuzzy output terms and need not sum to one. For an Iris antecedent associated with both *versicolor* and *virginica*, for example, soft correction may assign *medium* certainty to the former and *very large* certainty to the latter. This representation preserves competing consequents while allowing different degrees of support.

Detecting a conflict identifies an ambiguity in the rule base, but does not establish which class is correct. Both correction strategies therefore make knowledge-dependent revisions whose predictive effects are evaluated against the uncorrected setting.

3.4 Translation into Executable Fuzzy Rules

The resulting linguistic rules are translated into Python statements using few-shot prompts with two examples of the `scikit-fuzzy` control-rule syntax. The prompt specifies the feature variables, class outputs, and allowed linguistic terms, and requests code following a fixed output marker so that the generated statements can be extracted.

Each class has a separate fuzzy output variable on $[0, 1]$. A linguistic rule with antecedent $\mathbf{a}_k$ is represented by class-specific control rules

$$\begin{aligned} R_{k,c}: \quad &\text{IF} \bigwedge_{d=1}^{D} (\widetilde{x}_d \text{ is } a_{kd}), \\ &\text{THEN } z_c \text{ is } B_{k,c}, \end{aligned} \tag{7}$$

where $B_{k,c}$ is the fuzzy output term assigned to class c. For a rule with a single consequent, that class receives the *very large* output term and the remaining classes receive *very small*. For a softly corrected rule, candidate classes receive their assigned certainty terms and other classes receive *very small*. In the uncorrected setting, the original rules are translated separately, so conflicting consequents remain present during inference.

The code-generation stage expresses the rule base using predefined membership functions and control operations. It does not introduce a separate gradient-based training stage. The generated rules and their executable counterparts form intermediate artifacts that can be inspected before the classifier is used.

3.5 Prediction with Mamdani Fuzzy Inference

Let $\mu_{d,\ell}(v)$ be the membership of feature value v in linguistic term ℓ, and let K be the number of linguistic rules after the chosen correction procedure. For a normalized query $\widetilde{\mathbf{x}} = T(\mathbf{x})$, the standard min–max Mamdani formulation computes the activation of rule k as

$$\alpha_k(\widetilde{\mathbf{x}}) = \min_{d=1,\dots,D} \mu_{d,a_{kd}}(\widetilde{x}_d). \tag{8}$$

The activated consequent memberships are aggregated separately for each class:

$$\nu_c(z \mid \widetilde{\mathbf{x}}) = \max_{k=1,\dots,K} \min\left\{\alpha_k(\widetilde{\mathbf{x}}), \mu_{B_{k,c}}(z)\right\}, \qquad z \in [0,1]. \tag{9}$$

For a nonzero aggregated output, centroid defuzzification yields a class score:

$$s_c(\widetilde{\mathbf{x}}) = \frac{\int_0^1 z\, \nu_c(z \mid \widetilde{\mathbf{x}})\, \mathrm{d}z}{\int_0^1 \nu_c(z \mid \widetilde{\mathbf{x}})\, \mathrm{d}z}. \tag{10}$$

The predicted label is

$$\widehat{y} = f_{\mathcal{R}}(\widetilde{\mathbf{x}}) = \arg\max_{c \in \mathcal{Y}} s_c(\widetilde{\mathbf{x}}). \tag{11}$$

These scores summarize fuzzy support for each class; they are not calibrated class probabilities. All operations in Equations (8)–(11) are performed by the constructed fuzzy system. The role of the LLM is confined to offline rule construction, allowing the deployed decision procedure to be examined through its membership functions, activated rules, and class outputs.

4 Experiments

We evaluate whether the explicit fuzzy classifiers introduced in Section 3 provide effective tabular predictions, how rule correction and membership functions affect their behavior, and what computational cost is incurred during construction. The experiments separate predictive comparisons from analyses of the individual components.

4.1 Experimental Setup

Datasets and evaluation protocol. We use five numerical tabular classification datasets: Iris, Ecoli, Glass, Seeds, and User Knowledge Modeling (UKM). They cover flower classification, protein localization, glass identification, wheat variety classification, and student knowledge assessment, respectively. The numbers of features are 4, 7, 9, 7, and 5. Each dataset is divided approximately equally into a labeled rule-construction set and a test set, giving training/test sizes of $75/75$, $168/168$, $107/107$, $105/105$, and $201/202$, respectively. Numerical features are normalized to $[0, 1]$ before rule construction and fuzzy inference. We report classification accuracy in percent; differences between accuracies are expressed in percentage points. The supplied results record one accuracy per configuration. They do not establish run-to-run variability or statistical significance.

Fuzzy-LLM. The rule-construction experiments use DeepSeek-V3.2, Kimi K2, and Qwen3-235B as pretrained LLM backbones. Following Section 3.3, we examine no correction, hard correction, and soft correction. We also evaluate triangular, trapezoidal, and Gaussian membership functions. The linguistic vocabulary contains five terms with reference centers at 0, 0.25, 0.5, 0.75, and 1.

Table 1: Recorded test accuracy (%). Best direct LLM and best fine-tuned Qwen are dataset-wise maxima within those baseline groups. Fuzzy-LLM is likewise the best observed configuration per dataset (Table 2). These selections are exploratory. Bold marks the largest value in each column.

Method	Iris	Ecoli	Glass	Seeds	UKM
Best direct LLM	88.00	21.43	35.51	41.90	65.35
ANFIS	93.33	85.71	59.81	85.71	77.72
Random Forest	90.67	**86.90**	64.49	**95.24**	88.12
XGBoost	85.33	82.74	**68.22**	93.33	90.59
Best fine-tuned Qwen	90.67	63.69	60.75	86.67	**92.08**
Fuzzy-LLM	**94.67**	82.14	61.68	94.29	76.73

Table 2: One configuration attaining each reported Fuzzy-LLM maximum. Other configurations may tie; Supplementary Tables 6–10 list all recorded results.

Dataset	Backbone	Correction	Membership	Accuracy (%)
Iris	DeepSeek-V3.2	Soft	Gaussian	94.67
Ecoli	DeepSeek-V3.2	Soft	Gaussian	82.14
Glass	DeepSeek-V3.2	None	Gaussian	61.68
Seeds	Kimi K2	Hard	Trapezoidal	94.29
UKM	DeepSeek-V3.2	Soft	Gaussian	76.73

LLM parameters remain fixed, and predictions are obtained from the generated Mamdani fuzzy systems. Table 1 summarizes the recorded comparisons. Table 2 identifies one configuration attaining each Fuzzy-LLM maximum. The settings were identified from reported test accuracies, so these exploratory values do not estimate performance after independent validation-based selection.

Baselines. We consider three groups of reference methods. Direct LLM baselines predict class labels from textual descriptions of numerical features. Conventional supervised baselines comprise ANFIS, Random Forest, and XGBoost. Fine-tuned baselines comprise Qwen models with 4B, 8B, 14B, and 32B parameters, adapted using either LoRA or full fine-tuning. The direct-prediction group measures the performance of the reported prompting approach, whereas the supervised groups provide comparisons with methods that learn from labeled training examples. These groups represent different uses of pretrained knowledge and supervision.

4.2 Classification Results

Table 1 summarizes the recorded comparisons; Table 3 gives every baseline result, and Table 2 identifies a configuration attaining each Fuzzy-LLM maximum. The five selected Fuzzy-LLM accuracies are 94.67% on Iris, 82.14% on Ecoli, 61.68% on Glass, 94.29% on Seeds, and 76.73% on UKM. Because these settings were identified from the displayed test results, the comparison is exploratory rather than an estimate from independent validation-based model selection.

Fuzzy-LLM exceeds the strongest recorded direct LLM result by 6.67, 60.71, 26.17, 52.39, and 11.38 percentage points on Iris, Ecoli, Glass, Seeds, and UKM, respectively. This comparison changes both access to labeled examples and the prediction mechanism, so it does not isolate the benefit of fuzzy rules. Among supervised methods, Fuzzy-LLM leads on Iris and is within 0.95 points of Random Forest on Seeds; Random Forest or XGBoost performs better on Ecoli, Glass, and UKM. The best fine-tuned Qwen result reaches 92.08% on UKM, 15.35 points above Fuzzy-LLM. These are point estimates, with no repeated-run uncertainty supplied.

4.3 Effect of Rule Correction

Table 4 compares no correction, soft correction, and hard correction while fixing Gaussian membership functions. Each row holds the dataset and LLM backbone fixed, so changes within a row reflect the correction strategy.

Table 3: Complete recorded baseline results (accuracy, %). Fuzzy-LLM uses the best observed configuration per dataset. Bold marks the column maximum; LoRA and Full distinguish Qwen fine-tuning strategies.

Method	Iris	Ecoli	Glass	Seeds	UKM
Direct LLM prediction					
Baichuan4-Turbo	69.33	13.69	32.71	20.00	51.49
DeepSeek-V3	86.67	20.24	32.71	41.90	64.36
ERNIE-Speed-8K	46.67	0.60	30.84	7.62	31.19
GLM-4-Plus	88.00	17.86	34.58	34.29	65.35
Hunyuan-Turbo	40.00	5.95	31.78	33.33	52.48
MiniMax-Text-01	65.33	2.38	32.71	35.24	62.38
Qwen-Max	85.33	20.24	31.78	32.38	61.39
Spark Max	61.33	1.79	13.08	30.48	49.50
step-1-8k	70.67	21.43	35.51	31.43	55.94
Yi-Lightning	84.00	8.93	33.64	37.14	62.87
Conventional supervised classifiers					
ANFIS	93.33	85.71	59.81	85.71	77.72
Random Forest	90.67	**86.90**	64.49	**95.24**	88.12
XGBoost	85.33	82.74	**68.22**	93.33	90.59
Fine-tuned Qwen models					
Qwen-4B (LoRA)	50.67	42.86	35.51	36.19	34.16
Qwen-4B (Full)	66.67	42.86	38.32	35.24	**92.08**
Qwen-8B (LoRA)	65.33	42.86	32.71	33.33	46.04
Qwen-8B (Full)	84.00	42.86	46.73	33.33	**92.08**
Qwen-14B (LoRA)	68.00	42.86	32.71	33.33	55.45
Qwen-14B (Full)	90.67	42.86	35.51	35.24	40.59
Qwen-32B (LoRA)	69.33	42.86	35.51	33.33	55.94
Qwen-32B (Full)	86.67	63.69	60.75	86.67	**92.08**
Fuzzy-LLM (ours)	**94.67**	82.14	61.68	94.29	76.73

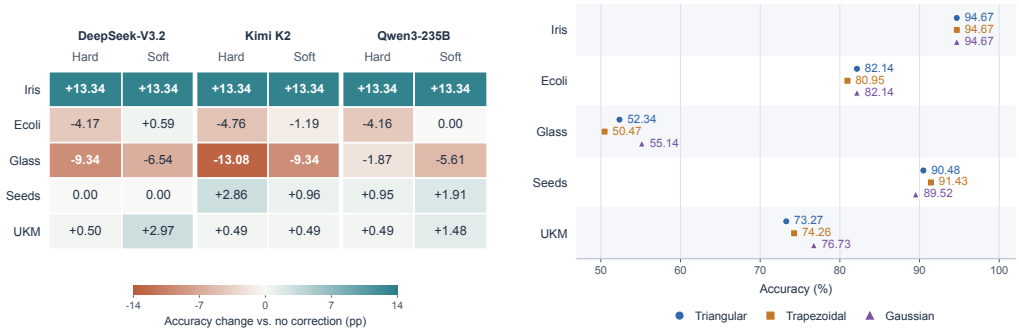
Table 4: Effect of rule correction with Gaussian membership functions. Values are accuracies (%); bold indicates the highest value within each row. None retains the original rules.

Dataset	Backbone	None	Soft	Hard
Iris	DeepSeek-V3.2	81.33	**94.67**	**94.67**
	Kimi K2	81.33	**94.67**	**94.67**
	Qwen3-235B	81.33	**94.67**	**94.67**
Ecoli	DeepSeek-V3.2	81.55	**82.14**	77.38
	Kimi K2	**80.95**	79.76	76.19
	Qwen3-235B	**80.95**	**80.95**	76.79
Glass	DeepSeek-V3.2	**61.68**	55.14	52.34
	Kimi K2	**61.68**	52.34	48.60
	Qwen3-235B	**61.68**	56.07	59.81
Seeds	DeepSeek-V3.2	**89.52**	**89.52**	**89.52**
	Kimi K2	89.52	90.48	**92.38**
	Qwen3-235B	88.57	**90.48**	89.52
UKM	DeepSeek-V3.2	73.76	**76.73**	74.26
	Kimi K2	73.27	**73.76**	**73.76**
	Qwen3-235B	73.27	**74.75**	73.76

On Iris, either correction strategy raises accuracy from 81.33% to 94.67% for all three backbones. The same values hold for triangular and trapezoidal membership functions. On Glass, however, both strategies lower accuracy for every backbone. Ecoli also shows mixed behavior: soft correction raises DeepSeek-V3.2 from 81.55% to 82.14%, leaves Qwen3-235B unchanged at 80.95%, and lowers Kimi K2 from 80.95% to 79.76%. Thus, a conflicting linguistic antecedent does not by itself show that LLM revision will improve classification.

Table 5: Membership-function analysis with DeepSeek-V3.2 and soft correction fixed. Values are accuracies (%); bold indicates the highest value in each row.

Dataset	Triangular	Trapezoidal	Gaussian
Iris	**94.67**	**94.67**	**94.67**
Ecoli	**82.14**	80.95	**82.14**
Glass	52.34	50.47	**55.14**
Seeds	90.48	**91.43**	89.52
UKM	73.27	74.26	**76.73**



(a) Effect of rule correction.

(b) Effect of membership functions.

Figure 2: Component analysis. (a) Percentage-point change from rule correction with Gaussian membership functions. (b) Accuracy across membership-function families with DeepSeek-V3.2 and soft correction.

4.4 Effect of Membership Functions

Table 5 fixes DeepSeek-V3.2 and soft correction to compare membership-function families. Iris remains at 94.67% for all three. Gaussian functions give the best UKM value in this setting (76.73%), whereas trapezoidal functions give the best Seeds value (91.43%). Triangular and Gaussian functions tie at 82.14% on Ecoli.

The dataset-wise maxima in Table 2 need not use this fixed DeepSeek-V3.2/soft-correction setting: Seeds reaches 94.29% with Kimi K2, hard correction, and trapezoidal functions. This distinction explains why the Seeds value in Table 5 is lower than the main comparison. The full grid appears in Supplementary Tables 6–10, and the graphical comparison appears in Figure 2.

4.5 Visual Analysis and an Inference Example

Figure 2 summarizes the component comparisons in Tables 4 and 5. The figure highlights the uniform gain from correction on Iris and its consistent loss on Glass. Figure 3 shows a held-out Iris sample with raw features $(6.1, 3.0, 4.9, 1.8)$. Rule R28 matches its four antecedents with degrees 1.000, 0.772, 0.728, and 0.923, giving activation 0.728. After all 29 rules are aggregated, the fuzzy class scores are 0.112 (setosa), 0.532 (versicolor), and 0.639 (virginica); the predicted class is virginica, matching the true label. These scores are not probabilities.

4.6 Offline Construction Cost

One recorded Iris construction case took about 13 s for rule generation, 2 s for soft correction, and 3 s for code generation, totaling about 18 s. Across a separate recorded workload covering the five datasets, $1{,}124$ API calls used an estimated $1{,}305{,}734$ input and $859{,}281$ output tokens. Under the pricing used in the recorded accounting, the estimated total cost is approximately US$1.40. The timing is a single case, and the cost does not cover every backbone and ablation run. Once constructed, the fuzzy classifier makes predictions without LLM calls.

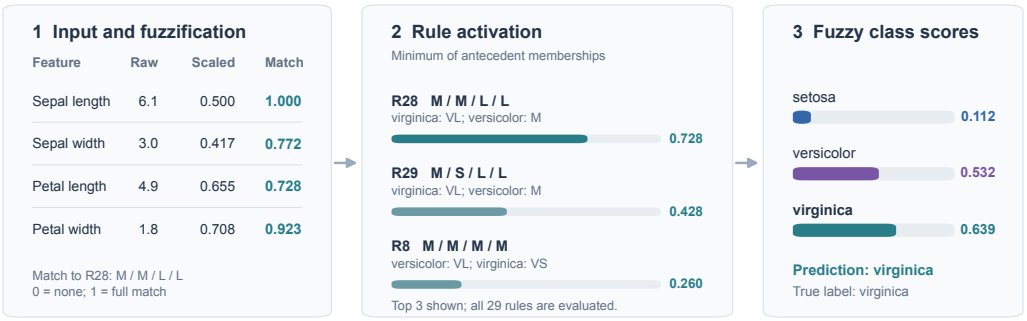


Figure 3: A held-out Iris prediction using DeepSeek-V3.2, soft correction, and Gaussian membership functions. Rule R28 activates at $\min(1.000, 0.772, 0.728, 0.923) = 0.728$. All 29 rules contribute to the final class scores, which are not calibrated probabilities.

4.7 Discussion

Fuzzy-LLM separates the source of the rules from the mechanism that executes them. For a new observation, the deployed classifier exposes normalized feature values, membership degrees, activated IF–THEN rules, and class scores (Figure 3). This makes a prediction traceable through the stored rule base and membership functions. The scores indicate fuzzy support rather than calibrated probabilities, and the trace does not explain why the LLM chose a particular rule during construction.

Rule correction behaves differently across tasks. With Gaussian membership functions, both correction methods increase Iris accuracy across all three backbones, but the original rules remain stronger on Glass. Distinct numerical observations can share a five-term linguistic antecedent even when their labels differ. Hard correction chooses one consequent, whereas soft correction keeps graded alternatives; neither treatment is guaranteed to preserve the useful information in a conflict. Membership-function choice also changes results, so rule construction and inference-time fuzzification should be considered together.

The dataset-wise maxima were identified from test accuracies, and the direct-LLM baselines do not use the same labeled examples. The reported advantage over prompting therefore describes the full workflow rather than the isolated effect of fuzzy rules. Conventional classifiers lead on Ecoli, Glass, and UKM. The LLM is invoked only during offline construction; the US$1.40 estimate covers a separate recorded workload rather than every configuration in the sweep. Validation-based selection, repeated runs, and a deterministic no-LLM rule constructor would help establish when LLM guidance adds predictive value.

5 Conclusion

Fuzzy-LLM offers a concrete bridge between connectionist language models and rule-based fuzzy reasoning for numerical tables. A pretrained LLM constructs linguistic IF–THEN rules offline, handles conflicting consequents using hard or soft correction, and translates the rules into a Mamdani classifier. New observations are classified without further LLM calls, while membership degrees, activated rules, and class scores make each decision inspectable. This division of labor uses the LLM to construct linguistic descriptions and leaves the numerical decision to explicit fuzzy operations. Across Iris, Ecoli, Glass, Seeds, and UKM, the best observed accuracies are 94.67%, 82.14%, 61.68%, 94.29%, and 76.73%, respectively, exceeding the recorded direct-prompting baselines on all five datasets. The component analyses show how rule correction and membership-function choice shape performance. Together, these results demonstrate an executable and interpretable way to connect language-guided rule construction with fuzzy inference.

AI USE STATEMENT

During the preparation of this work, the authors utilized generative AI tools, specifically Codex, for English language editing and text polishing to improve clarity and readability. Additionally, Codex was employed as a coding assistant to facilitate debugging and accelerate the implementation of experimental scripts. All AI-generated suggestions and code snippets were thoroughly reviewed, verified, and integrated by the authors. The authors retain full responsibility for the scientific integrity, technical accuracy, and final content of this paper.

ETHICS STATEMENT

This work focuses on fundamental advancements in machine learning algorithms and does not directly involve human subjects, sensitive personal data, or high-risk applications. The datasets used in our experiments are standard, publicly available benchmarks, and their utilization complies fully with their respective licenses. We do not foresee any direct negative societal impacts or ethical concerns resulting from this research.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the proposed method, model architecture, training procedure, hyperparameters, and evaluation protocol in the main text and supplementary material. All experiments use publicly available benchmark datasets, with preprocessing steps and data splits specified to enable replication. We report the implementation details used in our experiments. The code and configuration files will be made available upon publication.

REFERENCES

Micaela Lucia Bangerter, Giuseppe Fenza, Domenico Furno, Mariacristina Gallo, Vincenzo Loia, Claudio Stanzione, and Ilsun You. A hybrid framework integrating LLM and ANFIS for explainable fact-checking. *IEEE Transactions on Fuzzy Systems*, 33(12):4180–4189, 2025. doi: 10.1109/TFUZZ.2024.3431710.

Zhoujun Cheng, Tianbao Xie, Peng Shi, Chengzu Li, Rahul Nadkarni, Yushi Hu, Caiming Xiong, Dragomir Radev, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Tao Yu. Binding language models in symbolic languages. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=lH1PV42cbF.

Honghua Dong, Jiayuan Mao, Tian Lin, Chong Wang, Lihong Li, and Denny Zhou. Neural logic machines. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=B1xY-hRctX.

Yury Gorishniy, Akim Kotelnikov, and Artem Babenko. TabM: Advancing tabular deep learning with parameter-efficient ensembling. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/c1ba41c694834aeef91ae161711d4939-Abstract-Conference.html.

Sungwon Han, Jinsung Yoon, Sercan O Arik, and Tomas Pfister. Large language models can automatically engineer features for few-shot tabular learning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 17454–17479. PMLR, 2024. URL https://proceedings.mlr.press/v235/han24f.html.

Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. TabLLM: Few-shot classification of tabular data with large language models. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 5549–5581. PMLR, 2023. URL https://proceedings.mlr.press/v206/hegselmann23a.html.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=cp5PvcI6w8_.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=IkmD3fKBPQ.

Jyh-Shing Roger Jang. ANFIS: Adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(3):665–685, 1993. doi: 10.1109/21.256541.

Yiyuan Li, Shichao Sun, and Pengfei Liu. FRoG: Evaluating fuzzy reasoning of generalized quantifiers in LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 7239–7256. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.411. URL https://aclanthology.org/2024.emnlp-main.411/.

Jie Lu, Guangzhi Ma, and Guangquan Zhang. Fuzzy machine learning: A comprehensive framework and systematic review. *IEEE Transactions on Fuzzy Systems*, 32(7):3861–3878, 2024. doi: 10.1109/TFUZZ.2024.3387429.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-Refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/91edff07232fb1b55a505a9e9f6c0ff3-Abstract-Conference.html.

Ebrahim H. Mamdani and Sedrak Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, 7(1):1–13, 1975. doi: 10.1016/S0020-7373(75)80002-2.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Wang. Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 3806–3824. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-emnlp.248. URL https://aclanthology.org/2023.findings-emnlp.248/.

Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. Table meets LLM: Can large language models understand structured table data? a benchmark and empirical study. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pp. 645–654, 2024. doi: 10.1145/3616855.3635752.

Yifan Wang, Hisao Ishibuchi, Jihua Zhu, Yaxiong Wang, and Tao Dai. Unsupervised fuzzy neural network for image clustering. In *2021 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 1–6, 2021.

Yifan Wang, Hisao Ishibuchi, Meng Joo Er, and Jihua Zhu. Unsupervised multilayer fuzzy neural networks for image clustering. *Information Sciences*, 622:682–709, 2023. doi: 10.1016/j.ins.2022.11.149.

Yifan Wang, Hisao Ishibuchi, Witold Pedrycz, Jihua Zhu, Xiangyong Cao, and Jun Wang. Convolutional fuzzy neural networks with random weights for image classification. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 8(5):3279–3293, 2024. doi: 10.1109/TETCI.2024.3375019.

Yifan Wang, Witold Pedrycz, Hisao Ishibuchi, and Jihua Zhu. Fusion of explainable deep learning features using fuzzy integral in computer vision. *IEEE Transactions on Fuzzy Systems*, 33(1):156–167, 2025. doi: 10.1109/TFUZZ.2024.3434589.

Xiaoqian Wu, Yong-Lu Li, Jianhua Sun, and Cewu Lu. Symbol-LLM: Leverage language models for symbolic system in visual human activity reasoning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/5edb57c05c81d04beb716ef1d542fe9e-Abstract-Conference.html.

Lotfi A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965. doi: 10.1016/S0019-9958(65)90241-X.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023. URL https://arxiv.org/abs/2303.18223.

Yuanhang Zheng, Zeshui Xu, and Xinxin Wang. The fusion of deep learning and fuzzy systems: A state-of-the-art survey. *IEEE Transactions on Fuzzy Systems*, 30(8):2783–2799, 2022. doi: 10.1109/TFUZZ.2021.3062899.