Let the Fuzzy Rule Speak: Enhancing In-context Learning Debiasing with Interpretability

Anonymous ACL submission

Abstract

Large language models (LLMs) often struggle with balanced class accuracy in text classification tasks using in-context learning (ICL), hindering some practical uses due to user dissatisfaction or safety risks caused by misclassifications. Retraining LLMs to address root causes in data or model priors is neither easy nor cost-effective. This paper delves deeper into the class accuracy imbalance issue, identifying that it arises because certain classes consistently receive disproportionately high ICL probabilities, causing under-prediction and lower accuracy for others. More importantly, probability ranges affect the imbalance differently, allowing for precise, range-specific corrections. We introduce FuRud (**Fu**zzy **Ru**le Optimization-based **D**ebiasing), a method for sample-level class probability correction. FuRud tackles interpretability challenges by determining why certain classes need corrections and tailoring adjustments for each instance's class probabilities which is powered by fuzzy sets with triangular membership functions, transforming a class probability based on the range it belongs to. By solving a nonlinear integer programming problem with a labeled set of ICL class probabilities to minimize class accuracy bias (COBias) and maximize overall accuracy, each class selects an optimal correction function from 19 triangular membership functions without updating an LLM, and the selected functions correct test instances at inference. Across seven benchmark datasets, FuRud reduces COBias by over half (56%) and improves overall accuracy by 21% relatively, outperforming state-of-the-art debiasing methods.

1 Introduction

In-context learning (ICL) allows large language models (LLMs) to perform text classification tasks by prompting them with a few demonstrative examples. However, the class accuracies are often imbalanced due to biases in the training data or model priors. Addressing such imbalance while improving overall accuracy is a compelling frontier in the realm of *debiasing*. The skewness in the output space can be alleviated by inference-time corrections on ICL output logits or probabilities. For example, Lin and You (2024) explicitly targets mitigating class accuracy differences and quantify them as COBias, the averaged pairwise class accuracy differences, and learn class-level correction weights. While effective, the prior method corrects any instance with the same set of correction weights, lacking considerations in capturing per-sample, per-class nuances.

A direct cause of the imbalance is that ICL often yields specific ranges of probabilities to each output class. Some classes receive high probabilities for any input, while others may not. The consequence is that the latter is less frequently chosen than the former, resulting in lower accuracies. On the sample level, for all instances of a ground-truth class A, it is a general observation that instances with a low output probability in class A will have lower accuracy compared to those instances with a higher output probability in class A. The latter instances may not need as much amplification in class A output probability as the former instances. This suggests that sample-level customized correction should be enabled to accommodate different ICL probability ranges within a same output class.

Therefore, we aim for a sample-level correction method that interpretively amplifies or reduces different ranges of an output class's probabilities. In this work, we address the pressing need for enhanced understandings in how biased ICL predictions happen with the following research questions, and propose a per-sample, per-class correction method using fuzzy representation techniques.

RQ1: What is the interpretability challenge in correcting in-context learned representations? Given an N-class classification dataset, we denote the m-th instance's input prompt and class

as (x_m, y_m), where x_m consists of a task instruction, few-shot demonstrative examples, the input text and a question on its class. An LLM in-context learns output class probabilities $\boldsymbol{p}_m = (p_{m1}, \ldots, p_{mN})$ (normalized over N classes), then the prediction $\hat{y}_m$ is $\arg\max_i p_{mi}$. The $\boldsymbol{p}_m$ may need corrections in one or more of the classes, to reduce imbalance in class accuracy and improve overall accuracy. The interpretability challenges raised in this process can be specified as (1) detecting which classes need corrections, and (2) for each correction-needed class, applying range-specific amplifications/reductions.

RQ2: How can we achieve interpretable corrections with fuzzy rules? In short, we leverage membership functions to achieve interpretable corrections. For more backgrounds, interpretable machine learning systems need a human-readable subset of input attributes to generate the target (Jethani et al., 2021; Carvalho et al., 2019), so they often use fuzzy rules and fuzzy memberships, which provide interpretable quantifiers of given attributes (such as the size, *Small*), to learn these systems (Vernon et al., 2024; Vilone and Longo, 2020; Ishibuchi and Nojima, 2007). In classical fuzzy rule classification systems, input attributes are assigned to fuzzy sets to generate rules for pattern classification (Ishibuchi et al., 1999, 2005; Nojima and Ishibuchi, 2016; Rudziński, 2016; Gorzałczany and Rudziński, 2017). A fuzzy classification system contains multiple human-readable rules, which can be as simple as "1. If attribute Bare Nuclei is *Small* then the consequent (predicted) class is *Benign*.2....3. If attribute Uniformity of Cell Size is *not Small* then the consequent class is *Malignant*." (Gorzałczany and Rudziński, 2017). Here, *Small* and *not Small* are fuzzy sets, and their corresponding membership functions quantify the level of participation of each given attribute in the respective fuzzy set.

In this work, we leverage the range-wise transformation capabilities of membership functions for debiasing. A membership function is a curve that maps an input attribute to a fuzzy value between 0 and 1 (Zadeh, 1965). Viewing class probabilities as input attributes, we can use membership functions to adjust the probabilities, as long as the membership functions are selected under debiasing objectives. The key intuition is that a membership function can asymmetrically amplify or reduce different ranges of inputs. As such, a fuzzy rule based debiaser is applied to p_{mi}, denoted as $f_{A_i}(p_{mi})$, where A_i is a fuzzy set for class i, and its membership function f_{A_i} maps p_{mi} to a corrected $p'_{mi} := f_{A_i}(p_{mi})$. The debiaser can be viewed as a **single rule**:

$$\text{If } \underbrace{\text{class 1 is } A_1 \text{ and ... and class } N \text{ is } A_N}_{\text{Antecedent}}$$
$$\text{then } \underbrace{\text{predict } \arg\max_j f_{A_j}(p_{mj})}_{\text{Consequent}} \quad (1)$$

Our goal is to optimize the selection of membership functions towards mitigating COBias and improving overall accuracy. Specially, we include a *Don't Change* membership function that will keep a class unchanged. When a correction is needed, a piece of the triangular function is activated for evaluating the corrected probability based on the range that the input probability belongs to.

To this end, we propose FuRud, a **Fu**zzy **Ru**le Optimization based **D**ebiasing method, which leverages combinatorial optimization (Section 3). Optimized on a labeled set of few-shot ICL output class probabilities, each class in a downstream task selects a membership function from 19 triangular membership functions for correction, optimizing a multi-objective of COBias minimization and overall accuracy maximization. It achieves good improvements in accuracy and COBias with sample-level corrections, shown by experiments and analyses (Section 4) and discussions (Section 5). Figure 1 is an overview: an optimization set of ICL class probabilities and ground-truth answers are input to the multi-objective nonlinear integer programming model, which jointly selects optimal functions for each class. For a test instance, the learned membership functions correct the ICL class probabilities.

To highlight, the membership functions selected by FuRud enable sample-level correction and interpretability. FuRud identifies if an LLM in-context learns an accurate class probability for a given instance, namely, if *Don't Change* is selected, it means the LLM has learned accurate output representations for the class; otherwise, corrections are performed on a per-sample basis. Our source code will be released upon paper publication. Our contributions are:

- We propose a fuzzy rule based debiasing method (FuRud) for per-sample, per-class ICL output correction.

- We formulate a multi-objective nonlinear integer programming model that selects triangular

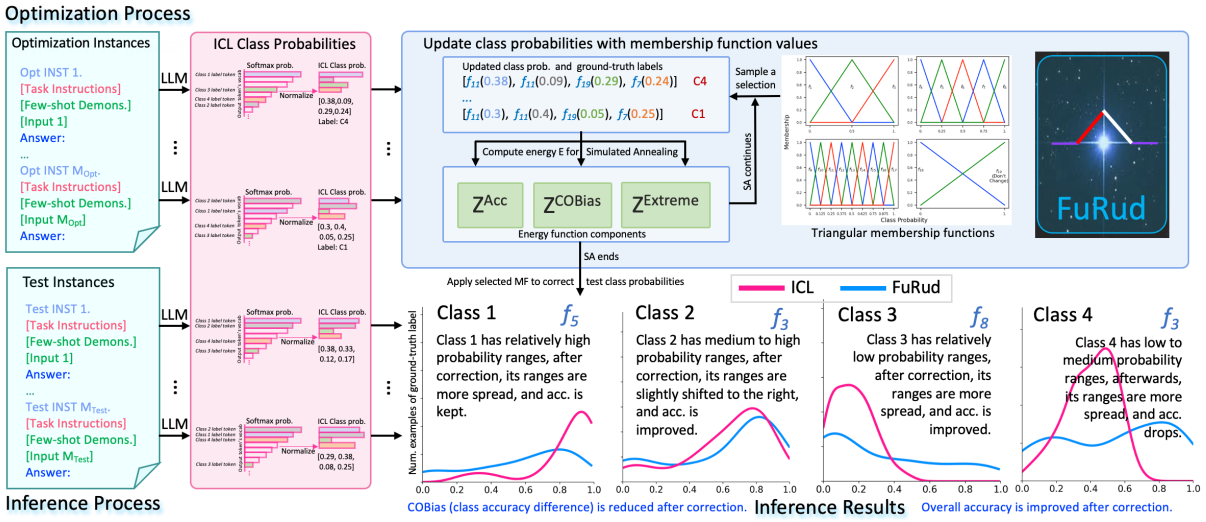

Figure 1: An overview of FuRud. ICL output probabilities across answer classes for input instances are obtained. On an optimization set, class probabilities and ground-truth labels are input to the FuRud multi-objective nonlinear integer programming model for joint learning of optimal membership functions. During inference, the optimal membership functions perform tailored corrections to class probabilities for test instances. This figure is for illustration purposes only, actual range changes and improvements are detailed in Section 4.

membership functions for each class, to minimize class accuracy differences (COBias) and maximize overall accuracy.

- Across seven benchmarks, FuRud has greatly reduced COBias and improved overall accuracy. For example, it reduces the origianl ICL COBias by a relative decrease of 56% and improves ICL accuracy by a relative increase of 21%; it also achieves higher accuracy (avg. accuracy reaching 72.0%) and competitive COBias (avg. COBias dropping to 17.8%) over state-of-the-art debiasing methods.

2 Related Work

Language Model Bias Mitigation. At the heart of debiasing is detecting biased patterns that arise in a large language model (LLM)'s outputs. Prior work has found various prediction biases in ICL, and address the biased patterns by methods of contextual prompt engineering and output adjustment (Brown et al., 2020; Schick et al., 2021; Zhao et al., 2021). Particularly, on classification tasks, researchers have found that LLMs' outputs are sensitive to ICL formatting, such as prompt templates, demonstrations, and verbalizers (Min et al., 2022; Holtzman et al., 2021; Schick and Schütze, 2021); besides, LLMs tend to output common tokens in the pre-training data (Zhao et al., 2021). These bias factors lead to majority label bias (Zhao et al., 2021), COBias (pairwise class accuracy differences) (Lin and You, 2024), *etc*, causing imbalanced per-class accuracies, and researchers address these biases by making output distribution calibrations (Zhao et al., 2021; Fei et al., 2023; Zhou et al., 2024), or by class probability re-weighting (Lin and You, 2024). For example, Zhao et al. (2021) calibrate the output distribution with content-free/dummy test prompts. Zhou et al. (2024) calibrate the output distribution in a test-time manner, estimating a contextual correction term of each class on a batch of test examples; the proposed Batch Calibration (BC) method outperforms previous calibration methods (Zhao et al., 2021; Fei et al., 2023) on a range of text classification tasks. Lin and You (2024) re-weights output probabilities by a set of class-specific weight coefficients; the proposed Debiasing as Nonlinear Integer Programming method (DNIP) achieves much lower COBias with higher accuracy than the ICL baseline. Though these debiasing methods effectively adjust ICL outputs, they do not emphasize interpretable bias handling. For example, a calibration method may not explicitly explain why a class needs corrections, or users may not fathom how a re-weighting method performs the exact corrections a class need.

3 Fuzzy Rule Optimization Based Debiasing

In the fuzzy rule setting, for N classes, each class selects a fuzzy set A_i, or equivalently, a member-

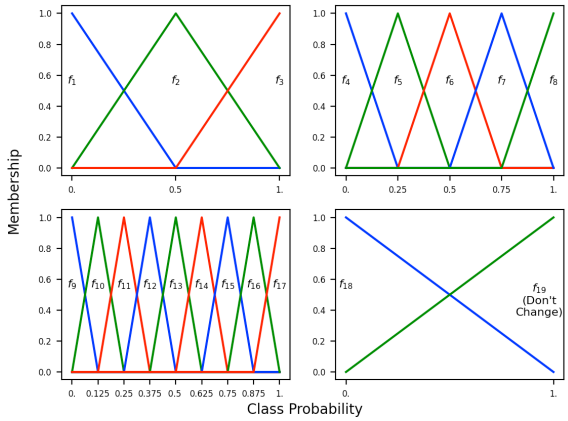

Figure 2: 19 triangular membership functions.

ship function f_{A_i}, from a family of K fixed fuzzy sets. We let $F = \{f_1, ..., f_k, ..., f_K\}$ denote the family of membership functions. The membership function selection problem is solved using simulated annealing. FuRud is a combinatorial optimization model, so it is performed in inference time on an optimization set of ICL output class probabilities with ground-truth labels, without LLM parameter updates. The selected membership functions are directly applied to transform test-time class probabilities.

Membership Functions. Each class selects from 19 triangular membership functions. Triangular membership functions are popular for fuzzy rule-based classification (Ishibuchi et al., 2005), as the speed of changes is easily controlled by the slope, and the linearity is computationally efficient. Without knowing the expected fuzzy partitions in downstream datasets, we employ four fuzzy partitions, resulting in 19 triangular membership functions of different granularities, shown in Figure 2, including the *Don't Change* membership function - the identity function. Other membership functions represent a sharp or smooth transformation of the input value. More details are provided in Appendix A. The general form of a triangular membership function $f_k(\cdot)$ can be written as:

$$f_k(p_{mi}; a_k, b_k, c_k) = \begin{cases} 0, & \text{if } p_{mi} \leq a_k \\ \dfrac{p_{mi} - a_k}{b_k - a_k}, & a_k \leq p_{mi} \leq b_k \\ \dfrac{c_k - p_{mi}}{c_k - b_k}, & b_k \leq p_{mi} \leq c_k \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where a_k, b_k, c_k are the left endpoint, the input value where the peak is reached, and the right endpoint of f_k. For example, for f_{11}, the a_k, b_k, c_k values are 0.125, 0.25, 0.375 respectively.

The updated probability p'_{mi} is computed by:

$$p'_{mi} = \begin{cases} p_{mi}, & \text{if } \sum_{i=1}^{N} p'_{mi} = 0 \\ \sum_k f_k(p_{mi}) \mathbb{1}(\kappa_i = k), & \text{otherwise} \end{cases} \tag{3}$$

where κ_i is the integer selection variable for class i. $\mathbb{1}(\cdot)$ evaluates to 1 if the condition inside is satisfied, otherwise 0. Furthermore, in case $p'_{mi} = 0$ for all classes, we reset each to be its original probability in $\boldsymbol{p}_m$. Therefore, $\hat{y}_m = \arg\max_i p'_{mi}$.

Multi-Objective Programming and Energy Function. Let $\boldsymbol{\kappa} = (\kappa_1, \ldots, \kappa_N)$ be the integer selection variables for classes $1, ..., N$, where κ_i is chosen from the given set of membership functions, and $\kappa_i = k$ means f_k is chosen. Our goal is to learn $\boldsymbol{\kappa}$ that improve ICL classifications under two main evaluation metrics, accuracy and COBias (Lin and You, 2024), i.e., our multi-objective spans across lowering COBias and increasing accuracy.

Crucially, we balance class accuracy by explicitly modeling COBias on the optimization set. The first objective is:

$$\min Z^{\text{COBias}} = \frac{1}{{}_N C_2} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \left| \text{Acc}_i - \text{Acc}_j \right| \tag{4}$$

where ${}_N C_2 = N(N-1)/2$, Acc_i is the accuracy score of class i on the optimization set.

The second objective improves overall accuracy:

$$\max Z^{\text{Acc}} = \frac{1}{|S^{\text{Opt}}|} \sum_{m \in S^{\text{Opt}}} \mathbb{1}\{\hat{y}_m = y_m\} \tag{5}$$

where S^{Opt} is the indices of optimization instances.

To further handle extreme cases of low class accuracies, we penalize classes that fail to reach an accuracy threshold, and minimize the loss between the threshold and per-class accuracy (cut off at 0). The third objective is:

$$\min Z^{\text{Extreme}} = \sum_{i=1}^{N} \max\{0, \lambda - \text{Acc}_i\} \tag{6}$$

where λ is a fixed threshold value.

The above objective functions are a mix of minimization and maximization, and the resulted multi-objective programming model requires integer variables. Each of them alone corresponds to an integer programming problem, which is NP-complete (Garey and Johnson, 1979). Classic solutions for

integer programming use operational research techniques, such as Branch-and-Bound, often used for linear integer programming problems. It could be difficult for such methods to handle nonlinear integer programming models which contain non-differentiable functions. Consequently, a series of metaheuristic algorithms have emerged, such as Simulated Annealing (SA), and each metaheuristic has their own strengths and limitations. We use one of the metaheuristics, SA, to tackle the proposed mathematical model. The SA implementation follows (Lin and You, 2024). Since it is difficult to solve each one as an individual optimization problem and force an optimal solution, our strategy is instead to compute a weighted sum of $1 - Z^{\text{Acc}}, Z^{\text{COBias}}, Z^{\text{Extreme}}$ as a single energy function E to be optimized. Hence, the multi-objectives are combined into a total minimization objective:

$$\min_{\kappa} E(\kappa; \lambda, \boldsymbol{p}') \tag{7}$$

where $E(\kappa; \lambda, \boldsymbol{p}') = \omega + \sum_{h \in S^{\text{Obj}}} \gamma^h Z^h$, S^{Obj} is the penalty (objective) functions, and ω, γ^hs are penalty parameters. SA optimizes on E to obtain an optimal selection of membership functions.

In summary, Eq. 4 targets minimizing COBias, Eq. 5 targets maximizing overall accuracy, and Eq. 6 targets maximizing per-class accuracy, which enforces it to meet a threshold; Eq. 7 combines the three objectives as a multi-objective function. Details on how Eq. 7 is optimized are described in experimental setups (Section 4.1).

4 Experiments

4.1 Experimental Setups

Evaluation Tasks and Evaluation Metrics. The proposed method is evaluated on a diverse range of text classification datasets, including AGNews (Zhang et al., 2015), a 4-class news topic classification; DBpedia (Auer et al., 2007), a 14-class ontology classification dataset derived from Wikipedia; SST-5 (Socher et al., 2013), a 5-class sentiment classification dataset; TREC (Voorhees and Tice, 2000; Li and Roth, 2002), a 6-class question classification dataset; RTE (Dagan et al., 2006), a binary entailment recognition dataset; and two biomedical domain-specific datasets, including DDI (Segura-Bedmar et al., 2013), a 5-class drug-drug interaction relation extraction dataset; PubMedQA (Jin et al., 2019), a 3-class biomedical question answering dataset. Each evaluation dataset is split into optimization/development/test sets. We follow (Lin and You, 2024) to preprocess the datasets. Evaluation metrics are accuracy and COBias.

FuRud Setups. The 19 triangular membership functions in Figure 2 form the base of selections for FuRud. We take the full or a subset of training instances from a downstream dataset to perform FuRud optimization. We prompt Llama-2-13B in 1-shot manner to obtain softmax probabilities at the output token over the entire vocabulary, which is then normalized over the classes. These ICL class probabilities and ground-truth labels are used to form the optimization set. The energy function we used is a special form of Equation 7 with $\omega = 1, \gamma^{\text{Acc}} = -1, \gamma^{\text{COBias}} = \alpha, \gamma^{\text{Extreme}} = \beta$. That is, the final multi-objective optimization function is $min_{\kappa} Z = 1 - Z^{\text{Acc}} + \alpha Z^{\text{COBias}} + \beta Z^{\text{Extreme}}$, where we learn κ_i for class $i = 1, \ldots, N$ on the optimization set. Each κ_i is selected from the given set of membership functions, and $\kappa_i = k$ denotes that membership function f_k is selected. At inference time, let $p = (p_1, \ldots, p_i, \ldots, p_N)$ be the ICL class probabilities of a test instance, then these probabilities are transformed by the learned membership functions, according to Eq. 3. The final corrected prediction is $\hat{y} = \arg\max_i f_{\kappa_i}(p_i)$.

The FuRud model Z is solved using simulated annealing (SA). The core step of SA is to sample a new solution $\kappa = (\kappa_1, \ldots, \kappa_N)$, e.g., (16, ..., 8), which is evaluated against Z. If the new Z is smaller, FuRud accepts the new solution; otherwise, it accepts the new solution with an acceptance probability $exp(-\Delta Z/T)$, where T is the temperature at the step. The values of α, β are tuned on the development set. Since we do not know an estimate for the expected threshold value λ in downstream tasks, we set it to 0.5 for simplicity. Prompting is done on a 80G A100 GPU. The SA algorithm executes on an AMD EPYC 7742 CPU in minutes.

We compare FuRud with the ICL baseline and two state-of-the-art ICL debiasing methods, including DNIP (Lin and You, 2024) and BC (Zhou et al., 2024). For fair comparisons, for each dataset, we prompt with three different 1-shot demonstrations and obtain three sets of initial probabilities. The demonstration is randomly sampled from optimization examples. The average test accuracy and COBias over three runs are reported.

4.2 Main Results

Table 1 shows the test accuracy and COBias of ICL, BC, DNIP, and FuRuD. Comparing FuRud to the ICL baseline, the average relative accuracy increase

Task	Acc. ↑				COBias ↓			
	ICL	BC	DNIP	FuRud	ICL	BC	DNIP	FuRud
AGNews	$79.9_{7.0}$	$82.5_{5.0}$	$87.9_{0.7}$	$85.7_{3.4}$	$28.3_{16.1}$	$23.1_{12.1}$	$6.3_{0.6}$	$6.9_{1.6}$
DBpedia	$88.6_{1.7}$	$89.1_{1.5}$	$93.4_{0.6}$	$92.2_{0.4}$	$16.2_{3.7}$	$15.4_{3.3}$	$7.7_{0.6}$	$9.2_{0.6}$
SST-5	$44.9_{4.3}$	$47.6_{2.3}$	$48.3_{1.9}$	$48.8_{3.8}$	$53.1_{5.0}$	$49.8_{10.7}$	$18.7_{10.1}$	$22.2_{8.4}$
TREC	$68.5_{10.8}$	$72.9_{4.4}$	$77.1_{2.0}$	$77.3_{3.9}$	$35.9_{6.5}$	$31.9_{5.1}$	$14.2_{1.3}$	$18.5_{1.4}$
RTE	$71.5_{2.2}$	$76.1_{0.6}$	$74.3_{0.8}$	$74.5_{1.8}$	$43.4_{7.0}$	$16.4_{1.9}$	$4.3_{3.3}$	$7.1_{5.0}$
DDI	$7.2_{0.9}$	$14.4_{2.5}$	$40.4_{6.0}$	$69.3_{6.3}$	$45.6_{5.9}$	$32.6_{7.6}$	$7.5_{3.2}$	$36.8_{4.6}$
PubMedaQA	$55.1_{2.9}$	$55.5_{1.3}$	$63.1_{14.0}$	$55.9_{5.4}$	$61.2_{1.9}$	$26.2_{3.2}$	$41.1_{29.6}$	$24.0_{8.4}$
Avg.	59.4	62.6	69.2	**72.0**	40.5	27.9	**14.3**	17.8

Table 1: Test accuracy and COBias (%); average scores over three runs are reported.

Dataset, Classes	Test Sentence (w/o prompt)	Test Label	ICL Class Prob.	ICL Prediction	Membership Function	Corrected Class Prob.	Corrected Prediction	Interpretations
AGNews World, Sports, Business, Tech	US unemployment claims slip but picture still murky, NEW YORK Fewer Americans lined up to claim first-time jobless benefits last week but analysts said the modest decline said very little about the current state of the labour market.	Business	[0.42, 0.01, 0.32, 0.25]	World	f_7, f_{16}, f_{11}, f_7	[0, 0, 0.47, 0]	Business	By FuRud, all classes need corrections. For this test instance, original ICL wrongly predicts Busi. as World. After correction, probability of class Busi. becomes highest, leading to the right prediction.
DBpedia Company, School, Artist, Athlete, Politician, Transportation, Building, Nature, Village, Animal, Plant, Album, Film, Book	Floyd Thomas Christian Sr. (December 18 1914 – May 11 1998) was Florida Commissioner of Education from 1965 to 1973. Christian was born in Bessemer Alabama. He moved to Pinellas County with his family in 1927...	Politician	[0, 0.16, 0.08, 0.64, 0.12, 0, 0, 0, 0, 0, 0, 0]	Athlete	f_3, f_{16}, f_7, f_8, DC, $f_2, f_2, f_2, f_{16}, f_2, f_{16}$, DC, DC, DC	[0, 0, 0, 0, 0.12, 0, 0, 0, 0, 0, 0, 0]	Politician	By FuRud, 4 out of 14 classes apply Don't Change (Their ICL probability is relatively accurate.), including class Politician. Though ICL probability of the actual class Politic. is 0.12 and unchanged after correction, classes like Ath.'s probability is corrected to 0 by f8, leading to the right prediction.

Table 2: Examples of sample-level corrections and explanations.

is 21%, and the average relative COBias reduction is 56%. The average test accuracy of FuRud over seven benchmarks is 72%, which outperforms the accuracy of BC and DNIP; the average test COBias of FuRud is 17.8%, which is comparable to DNIP with obtains the lowest COBias (14.3%) among the methods compared. It is noted that FuRud uses the full optimization set to make a fair comparison to DNIP. However, FuRud can also work in a few-shot optimization manner, as discussed in Appendix B. On top of that, FuRud provides per-sample, per-class interpretability, analyzed as follows.

4.3 Interpretability Analysis

Across seven tasks, AGNews is the only task that *Don't Change* was not applied to any class over any of its three runs with different initial ICL probabilities; RTE, DDI, and PubMedQA applied *DC* to a single class at most; TREC and SST-5 applied *DC* to a two classes at most; DBpedia, with 14 classes, at most applied *DC* to 4 classes, showing that most ICL output classes need correction.

In addition, Figure 3 visualizes range-specific probability changes after applying membership function corrections on AGNews and RTE, demonstrating that the membership functions selected by FuRud effectively transform different probability ranges of each output class to improve or at least maintain class accuracy, while making class accuracies more balanced. Moreover, the worst-performing class by ICL in either task is significantly improved by FuRud, due to that lower and medium probability ranges of the worst-performing class gets amplified and the higher ranges of other classes gets slightly reduced. Results on other datasets are similar. Table 2 exemplifies how per-sample, per-class corrections are interpreted.

We further quantitatively evaluate the ratio of instances that benefit from the correction. Figure 4 shows the accuracy of range-specific instances within class *Business* of AGNews. This class has 1,204 test instances, which are divided into 5 groups ranging from $[0.0, 0.2]$ to $[0.8, 1.0]$ based on their initial ICL output probability in *Business*. For example, 108 instances have ICL *Business* probabilities in $[0.2, 0.4]$ and only 9% of these instances are correctly predicted. This group of instances is effectively corrected by membership function f_{11} and synergetic corrections in other classes, reaching 66% accuracy after correction, i.e., 57% (66%-9%) more instances in this group obtain the right predictions after corrections.

5 Discussion

5.1 FuRud on Letter Based ICL Outputs

FuRud greatly improves highly skewed letter based ICL Outputs. In details, we experiment with the letter answer prompts, which is widely used in clas-

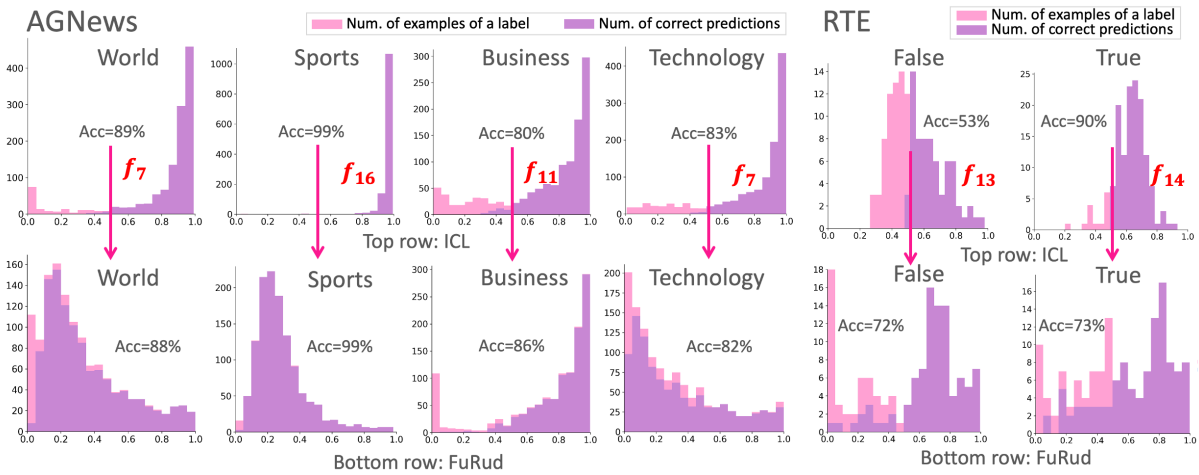

Figure 3: Class probabilities before and after applying corrections. For each task, we report results of the seed 1 run out of 3 runs. There was a stark ICL accuracy difference of 37% between *True* and *False* on RTE. FuRud addresses it by amplifying the medium range of *False* and simultaneously reducing the relatively high range of *True*.

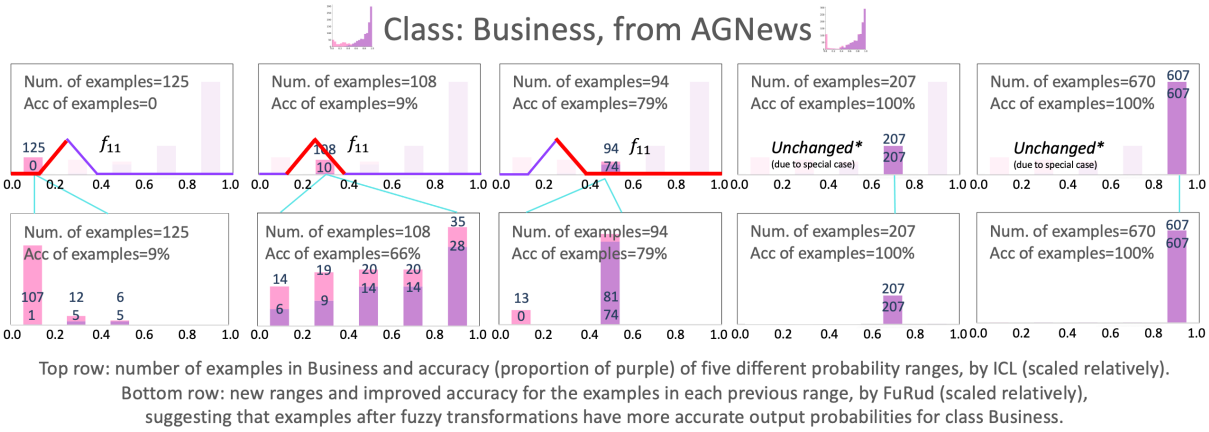

Figure 4: Quantitative evaluation on the ratio of instances that benefit from the correction, exemplified by class *Business* of AGNews. The difference of "Acc of examples" between bottom and top subfigures represents the ratio. The red color highlights the activated pieces of the membership function for range-specific correction.

sification and reasoning tasks. Letter options could lead to more shallow pattern matching problems than using label token as answer options. Using this prompt, the model outputs a single letter choice of "A", "B", *etc.* corresponding to a class label, which often results in highly skewed outputs, because LLMs have a tendency to select a certain letter option regardless of the content (Bentham et al., 2024). We find similar issues when evaluating seven datasets using letter options. For example, on AGNews, the LLM biases to predict "B" (*Sports*), leading to an average of 99% accuracy in *Sports* and 12% accuracy in *Business*. FuRud improves accuracy by an relative 44% and achieves a significant COBias reduction of a relative 54% over ICL, as shown in Table 3 (averaged over seven datasets).

Method	Acc.	COBias
ICL (letter)	$36.9_{13.6}$	$47.2_{15.6}$
FuRud (letter)	$53.1_{10.5}$	$21.6_{8.2}$

Table 3: Letter based results.

Besides the tabled results, on AGNews, overall test accuracy improves from 45% to 66%, COBias reduces from 54% to 10%. Class accuracy changes are: *World*, 40% → 69%; *Sports*, 99% → 70%; *Business*, 12% → 66%; *Technology*, 27% → 59%. These results suggest that FuRud can debias ICL class probabilities no matter if an input prompt leads to spurious label matching results.

5.2 Membership Function Granularities

We experiment with different combinations of the four fuzzy partitions in Figure 2 and show that membership granularities lead to accuracy-COBias

tradeoffs. In details, we conduct five ablations based on the four partitions characterized by different slopes $\pm1, \pm2, \pm4, \pm8$, where a bigger slope indicates higher granularities. The ±1 partition is the DC partition (f_{18}, f_{19}). Since it plays a vital role in maintaining some classes, we keep it in all five combinations, including DC alone, DC and each of the rest partitions, and mixed partitions (all four partitions). The average scores across seven datasets are reported; for each dataset, average accuracy and COBias over three runs with different demonstrations is taken.

As shown by Figure 5, while COBias greatly reduces with higher membership granularities, overall accuracy slightly decreases.

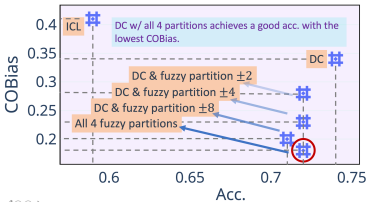

Figure 5: Accuracy-COBias tradeoff of fuzzy partitions.

Therefore, although it is tempting to include more fine-grained membership functions to reduce COBias, don't forget the accuracy-COBias tradeoff. The best accuracy and COBias is achieved with mixed partitions. Moreover, although he DC partition alone can obtain 15% higher accuracy than ICL accuracy, but the improvement mainly comes from a single task (DDI). In addition, we added a side analysis with partition ±8 alone, while achieving similar accuracies, the COBias is 6% higher than using DC and partition ±8 together, suggesting that the *Don't Change* function is essentially needed when using mixed partitions to attain both good COBias and overall accuracy.

5.3 Additional Analyses

More Models. On varied relatively small LLMs, FuRud consistently improves their performances, showcased by additional results on Llama-2-7B and GPT-2-XL in Appendix C.

More ICL Strategies. FuRud significantly improves accuracy and COBias for a more sophisticated prompting case, where the demonstration in the prompt uses N-shot examples taken from each class, as detailed in Appendix D.

Computational Costs. The computational cost of FuRud is low. Execution time of the optimization program ranges rom several minutes to around 30 minutes, depending on the number of classes, optimization set sizes, and etc.

Interpretability Comparisons: FuRud vs DNIP. DNIP does not capture sample-level nuances needed in the correction as FuRud does, not suitable for classes that need fine-grained sample-level correction. The membership functions overcome this issue, explaining how each class in a particular instance should be corrected, and this is the main innovation of our paper.

Using traditional fuzzy classification systems. Training involves extensive computations as it requires maintaining multiple candidate rules for debiasing, e.g., "R_q: If class 1 probability is A_{q1} and ... and class N probability is A_{qN}, then predict Y_q", and test time for a winning rule grows with the number of candidates. To obtain high accuracy, a huge number of rules may be employed, making the system inefficient. In contrast, FuRud is cost-effective, and implicitly reflects the many rules employed in a traditional system. For more discussions, please refer to Appendix E.

6 Conclusion

We present FuRud, a post-hoc debiasing method for sample-level ICL output correction that effectively enhances overall accuracy and accuracy balance across multiple classes, leveraging combinatorial optimization to select optimal fuzzy membership functions for interpretable, range-specific corrections on the original ICL class probabilities. On a diverse set of text classification benchmarks, FuRud greatly reduces COBias while enhancing overall accuracy over ICL results, outperforming state-of-the-art debiasing methods.

Limitations

There is a possibility that certain classes in a task may not need as much fine-grained sample-level correction as other classes. Those classes may apply class-level correction to any instance, e.g., weight coefficients, to obtain sufficient COBias reduction while being interpretable at the broader level. As such, an organic integration of both broad and fine-grained corrections should be studied in the future. In addition, class accuracy imbalances widely exist in variants of small (e.g., Llama-2-13B) and larger (e.g., ChatGPT) LLMs (Lin and You, 2024), how much FuRud could balance larger LLMs could be quantitatively analyzed to exemplify more practical usages of the method. This paper primarily focuses on a single small model for its representativeness as a widely applied LLM.

References

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. DBpedia: A Nucleus for A Web of Open Data. In *Proceedings of the 6th International The Semantic Web and 2nd Asian Conference on Asian Semantic Web Conference*, pages 722–735.

Oliver Bentham, Nathan Stringham, and Ana Marasović. 2024. Chain-of-thought unfaithfulness as disguised accuracy. *Preprint*, arXiv:2402.14897.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Diogo V. Carvalho, Eduardo M. Pereira, and Jaime S. Cardoso. 2019. Machine learning interpretability: A survey on methods and metrics. *Electronics*, 8(8).

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The PASCAL Recognising Textual Entailment Challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190.

Yu Fei, Yifan Hou, Zeming Chen, and Antoine Bosselut. 2023. Mitigating label biases for in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14014–14031.

M.R. Garey and D.S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-completeness*. Mathematical Sciences Series. Freeman.

Marian B. Gorzałczany and Filip Rudziński. 2017. Interpretable and accurate medical data classification – a multi-objective genetic-fuzzy optimization approach. *Expert Systems with Applications*, 71:26–39.

Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. 2021. Surface Form Competition: Why the Highest Probability Answer Isn't Always Right. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7038–7051.

Hisao Ishibuchi, Tomoharu Nakashima, and Tadahiko Murata. 1999. Performance Evaluation of Fuzzy Classifier Systems for Multidimensional Pattern Classification Problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 29(5):601–618.

Hisao Ishibuchi and Yusuke Nojima. 2007. Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning. *International Journal of Approximate Reasoning*, 44(1):4–31.

Hisao Ishibuchi, Takashi Yamamoto, and Tomoharu Nakashima. 2005. Hybridization of Fuzzy GBML Approaches for Pattern Classification Problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 35(2):359–365.

Neil Jethani, Mukund Sudarshan, Yindalon Aphinyanaphongs, and Rajesh Ranganath. 2021. Have We Learned to Explain?: How Interpretability Methods Can Learn to Encode Predictions in their Interpretations. *Proceedings of Machine Learning Research*, 130:1459–1467.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. PubMedQA: A Dataset for Biomedical Research Question Answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2567–2577.

Xin Li and Dan Roth. 2002. Learning Question Classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Ruixi Lin and Yang You. 2024. Cobias and debias: Minimizing language model pairwise accuracy bias via nonlinear integer programming. *Preprint*, arXiv:2405.07623.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the Role of Demonstrations: What Makes In-Context Learning Work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064.

Yusuke Nojima and Hisao Ishibuchi. 2016. Multiobjective Fuzzy Genetics-based Machine Learning with a Reject Option. In *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1405–1412.

Filip Rudziński. 2016. A multi-objective genetic optimization of interpretability-oriented fuzzy rule-based classifiers. *Applied Soft Computing*, 38:118–133.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online.

Timo Schick, Sahana Udupa, and Hinrich Schütze. 2021. Self-Diagnosis and Self-Debiasing: A Proposal for Reducing Corpus-Based Bias in NLP. *Transactions of the Association for Computational Linguistics*, 9:1408–1424.

Isabel Segura-Bedmar, Paloma Martínez, and María Herrero-Zazo. 2013. SemEval-2013 Task 9 : Extraction of Drug-Drug Interactions from Biomedical Texts (DDIExtraction 2013). In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 341–350.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642.

Eric M. Vernon, Naoki Masuyama, and Yusuke Nojima. 2024. Integrating white and black box techniques for interpretable machine learning. In *Proceedings of Ninth International Congress on Information and Communication Technology*, pages 639–649.

Giulia Vilone and Luca Longo. 2020. Explainable artificial intelligence: a systematic review. *Preprint*, arXiv:2006.00093.

Ellen M. Voorhees and Dawn M. Tice. 2000. Building a question answering test collection. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 200–207.

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

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level Convolutional Networks for Text Classification. In *Advances in Neural Information Processing Systems*.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate Before Use: Improving Few-shot Performance of Language Models. In *Proceedings of the 38th International Conference on Machine Learning*, pages 12697–12706.

Han Zhou, Xingchen Wan, Lev Proleev, Diana Mincu, Jilin Chen, Katherine A Heller, and Subhrajit Roy. 2024. Batch Calibration: Rethinking Calibration for In-Context Learning and Prompt Engineering. In *The Twelfth International Conference on Learning Representations*.

A Details on Membership Functions

Table 4 lists the details about the membership functions used in this work.

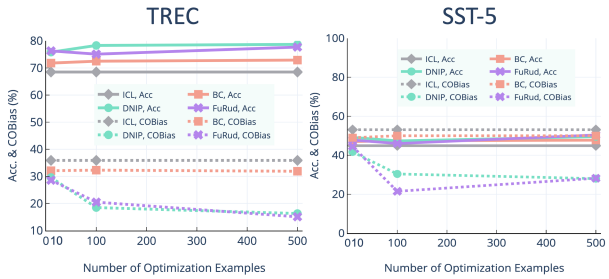

Figure 6: Few-shot optimization.

B Few-shot Optimization

FuRud can optimize a downstream task with as few as 10 examples. We take few-shot optimized TREC and SST-5 results for illustration. Figure 6 shows test accuracy and COBias of FuRud (in mint green color) when used in a few-shot optimization manner, starting with 10 few-shot examples and growing to 100 and 500 examples. TREC and SST-5 are shown to illustrate that FuRud can achieve an average of 9% accuracy improvements with 18% COBias reduction over the ICL baseline at 10 few-shot optimization examples.

At 10 examples, FuRud obtains a 11% and 6% relative increase in accuracy over the ICL baseline on TREC and SST-5 respectively, at the same time, it reduces COBias by a relative 20% and 16% on each dataset. The accruacy and COBias performances gradually improve as the number of examples increases to 500. Compared to existing methods, FuRud outperforms BC in few-shot scenarios, and performs better than (TREC) or on par (SST-5) with DNIP while being interpretable. Similar findings apply to the other five datasets, as shown in Figure 7. In short, FuRud achieves better or comparable results than DNIP, and better results than BC and the ICL baseline, while providing enhanced interpretability.

C FuRud's Performances on More LLMs

We ran experiments of FuRud on two additional models, Llama-2-7B and GPT2-XL. Results are shown in Table 5. For example, on Llama-2-7B, FuRud improves accuracy by a relative 22%, and reduces COBias by a relative 63% over ICL baselines, demonstrating that FuRud gains consistent performance improvements on various models. Indeed, our current evaluations are focused on relatively small LLMs, but our approach can also work for larger models, as long as class probabilities are available and the imbalanced per-class accuracy

Function	Parameters	Name	Short Form	Meaning
f_1	0, 0, 0.5	Low-2	L-2	Low-range transformation, smooth change with slope -2, peak at 0
f_2	0, 0.5, 1	Medium-2	M-2	Medium-range transformation, smooth change with slope ± 2, peak at 0.5
f_3	0.5, 1, 1	High-2	H-2	High-range transformation, smooth change with slope 2, peak at 1
f_4	0, 0, 0.25	Low-4	L-4	Low-range transformation, sharp change with slope -4, peak at 0
f_5	0, 0.25, 0.5	Medium Low-4	ML-4	Low-to-medium-range transformation, sharp change with slope ± 4, peak at 0.25
f_6	0.25, 0.5, 0.75	Medium-4	M-4	Medium-range transformation, sharp change with slope ± 4, peak at 0.5
f_7	0.5, 0.75, 1	Medium High-4	MH-4	Medium-to-high-range transformation, sharp change with slope ± 4, peak at 0.75
f_8	0.75, 1, 1	High-4	H-4	High-range transformation, sharp change with slope 4, peak at 1
f_9	0, 0, 0.125	Very Very Low-8	VVL-8	Very-very-low-range transformation, very sharp change with slope -8, peak at 0
f_{10}	0, 0.125, 0.25	Very Low-8	VL-8	Very-low-range transformation, very sharp change with slope ± 8, peak at 0.125
f_{11}	0.125, 0.25, 0.375	Low-8	L-8	Low-range transformation, very sharp change with slope ± 8, peak at 0.25
f_{12}	0.25, 0.375, 0.5	Medium Low-8	ML-8	Low-to-medium-range transformation, very sharp change with slope ± 8, peak at 0.375
f_{13}	0.375, 0.5, 0.625	Medium-8	M-8	Medium-range transformation, very sharp change with slope ± 8, peak at 0.5
f_{14}	0.5, 0.625, 0.75	Medium High-8	MH-8	Medium-to-high-range transformation, very sharp change with slope ± 8, peak at 0.625
f_{15}	0.625, 0.75, 0.875	High-8	H-8	High-range transformation, very sharp change with slope ± 8, peak at 0.75
f_{16}	0.75, 0.875, 1	Very High-8	VH-8	Very-high-range transformation, very sharp change with slope ± 8, peak at 0.875
f_{17}	0.875, 1, 1	Very Very High-8	VVH-8	Very-very-high-range transformation, very sharp change with slope 8, peak at 1
f_{18}	0, 0, 1	Full-1	F-1	Full-range transformation, very smooth change with slope -1, peak at 0
f_{19}	0, 1, 1	Don't Change	Don't Change	Identity function

Table 4: Names, parameters (a, b, c), short forms, and meanings for membership functions.

issue exists.

D FuRud's Performances under More ICL Demonstration Selection Strategies

We additionally prompt Llama-2-13B with the following demonstration selection strategy: k-shot prompting, where k is the number of classes. A demonstrative example from each class is randomly selected from the optimization set and represented in the prompt. FuRud significantly improves accuracy and COBias over ICL baselines, as shown in Table 6.

Compared to the 1-shot strategy (Table 1), the k-shot strategy provides a different starting point for FuRud. For example, the average ICL accuracy by k-shot (61.9%) is slightly larger than that obtained by 1-shot (59.4%), and average COBias (25.6%) is smaller than 1-shot (40.5%). FuRud boosts average accuracy to 73.5% and reduces COBias to 13.0%. In conclusion, different example selection strategies provide different starting points to optimize, on which FuRud consistently improve.

E More Discussions

We have a different motivation from traditional post-hoc corrections. Some may argue that ensuring equitable accuracies across all classes is a well-studied problem in standard machine learning classifiers. It is worth emphasizing that the per-class prediction accuracy imbalance should be treated within their particular context. The accuracy bias in ICL outputs stems from completely different causes than the unequal class accuracies observed in potentially overfitted traditional classifiers, where the former is rooted in prompts and the LLMs, and the latter arises from class imbalance of supervised training data. That's why our method is particularly applied to ICL's output token class probabilities, pinpointing specific patterns and ap-

Model	Metric	AGNews	DBpedia	SST-5	TREC	RTE	DDI	PubMedQA	Avg.
					Llama-2-7B				
ICL	Acc	$86.4_{2.5}$	$88.9_{2.0}$	$42.1_{11.1}$	$66.7_{6.6}$	$66.3_{4.3}$	$6.7_{0.4}$	$40.3_{6.7}$	56.8
	COBias	$14.0_{6.5}$	$13.5_{2.1}$	$55.6_{1.5}$	$33.2_{10.0}$	$61.6_{10.5}$	$41.4_{1.7}$	$40.9_{16.1}$	37.2
FuRud	Acc	$\mathbf{88.5_{0.5}}$	$\mathbf{91.5_{0.5}}$	$\mathbf{49.5_{0.7}}$	$\mathbf{73.1_{3.9}}$	$\mathbf{72.7_{1.0}}$	$\mathbf{54.4_{6.4}}$	$\mathbf{55.7_{7.6}}$	**69.3**
	COBias	$\mathbf{7.4_{2.5}}$	$\mathbf{8.4_{0.6}}$	$\mathbf{24.0_{1.2}}$	$\mathbf{14.1_{1.9}}$	$\mathbf{4.2_{2.7}}$	$\mathbf{16.9_{5.0}}$	$\mathbf{21.8_{16.6}}$	**13.8**
					GPT2-XL				
ICL	Acc	$52.1_{5.4}$	$31.8_{9.9}$	$34.9_{13.7}$	$27.4_{10.5}$	$55.4_{1.9}$	$14.5_{4.4}$	$55.2_{0.0}$	38.8
	COBias	$35.5_{11.5}$	$40.0_{3.6}$	$48.7_{5.4}$	$45.6_{8.7}$	$82.4_{24.5}$	$40.7_{5.9}$	$59.4_{12.6}$	50.3
FuRud	Acc	$\mathbf{69.0_{0.5}}$	$\mathbf{67.7_{11.8}}$	$\mathbf{43.4_{3.1}}$	$\mathbf{41.7_{2.7}}$	$\mathbf{51.2_{3.7}}$	$\mathbf{53.2_{17.0}}$	$\mathbf{48.4_{0.3}}$	**53.5**
	COBias	$\mathbf{7.4_{2.9}}$	$\mathbf{23.0_{6.5}}$	$\mathbf{25.4_{1.4}}$	$\mathbf{30.2_{7.0}}$	$\mathbf{8.9_{3.6}}$	$\mathbf{23.1_{6.5}}$	$\mathbf{17.6_{4.6}}$	**19.4**

Table 5: Test accuracy and COBias Comparisons on more LLMs.

Demonstration Selection	Metric	AGNews	DBpedia	SST-5	TREC	RTE	DDI	PubMedQA	Avg.
k-shot ICL	Acc	$83.5_{1.5}$	$95.2_{1.2}$	$50.3_{2.3}$	$67.0_{12.7}$	$75.0_{0.8}$	$9.7_{1.0}$	$52.3_{5.3}$	61.9
	COBias	$14.9_{5.1}$	$7.0_{2.2}$	$36.3_{7.2}$	$38.2_{5.1}$	$22.5_{13.2}$	$39.7_{3.5}$	$20.9_{4.2}$	25.6
FuRud	Acc	$\mathbf{88.1_{0.6}}$	$\mathbf{96.6_{0.4}}$	$\mathbf{54.3_{1.3}}$	$\mathbf{77.9_{6.0}}$	$\mathbf{75.9_{4.6}}$	$\mathbf{62.3_{2.1}}$	$\mathbf{59.2_{5.9}}$	**73.5**
	COBias	$\mathbf{7.7_{2.5}}$	$\mathbf{4.4_{0.7}}$	$\mathbf{13.8_{4.1}}$	$\mathbf{11.6_{3.3}}$	$\mathbf{5.0_{1.4}}$	$\mathbf{27.0_{2.2}}$	$\mathbf{21.3_{8.7}}$	**13.0**

Table 6: Test accuracy and COBias under the k-shot demonstration selection strategy.

plying precise, targeted corrections.

In the future, more versatile rules can be explored, and we may also examine the tradeoff between the accuracy and rule complexity. Simpler rules are easier to understand, but the transformations may fail to catch the intricate interactions between class predictions. More complex rules may have better modeling capabilities, but they are harder to read. In addition, this work focuses on evaluating text classification, and we will extend interpretable ICL debiasing to more language tasks, modalities, and model architectures.

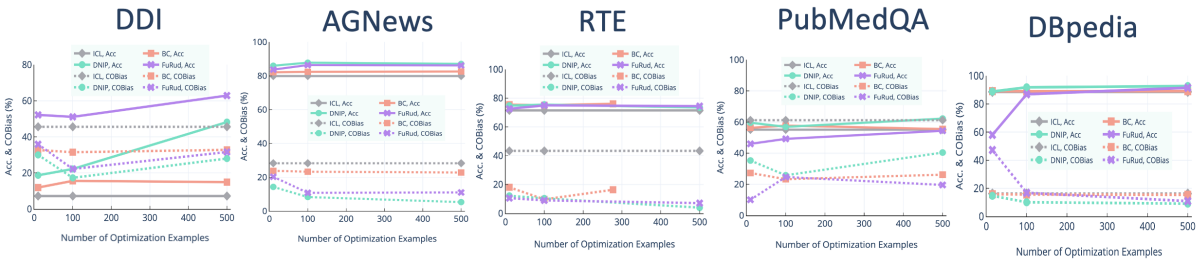

Figure 7: Additional few-shot optimization results.