Fuzzy Learning Machine

Junbiao Cui[1]
945546899@qq.com

Jiye Liang[1*]
ljy@sxu.edu.cn

[1] Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, School of Computer and Information Technology, Shanxi University, Taiyuan 030006, Shanxi, China.

Abstract

Classification is one of the most important problems in machine learning and the nature of it is concept cognition. So far, dozens of different classifiers have been designed. Although their working mechanisms vary widely, few of them fully consider concept cognition. In this paper, a new learning machine, fuzzy learning machine (FLM), is proposed from the perspective of concept cognition. Inspired by cognitive science, its working mechanism is of strong interpretability. At the same time, FLM roots in set theory and fuzzy set theory, so FLM has a solid mathematical foundation. The systematic experimental results on a large number of data sets show that FLM can achieve excellent performance, even with the simple implementation.

1 Introduction

Given an input space $\mathcal{X}$, an output space (a finite set) $\mathcal{Y}$, and an unknown function $\varphi : \mathcal{X} \to \mathcal{Y}$, the goal of classification is finding the φ or an approximation to it. Classification is one of the most important problems in machine learning, dating back to the origins of machine learning. In classification, each member of the output space $\mathcal{Y}$ corresponds to a concept. In essence, the process of classification is the process of concept cognition. Concept contains our knowledge about the world, and we use concept to understand and organize the world. Without it, there will be no human intelligence at all. That is why classification plays an important role in machine learning even in artificial intelligence.

Over the past decades, a large number of classifiers have been designed and achieved success on different tasks. The literature [1] systematically summarizes 179 classifiers from 17 different families, which contains almost all existing classifiers. Usually, generalization performance is regarded as an important dimension for evaluating classifiers. With the increasing depth and breadth of machine learning applications, classifiers are expected to possess more and more good properties, such as interpretability [2, 3, 4], robustness [5, 6, 7], and so on.

The existing classifiers follow different design principles (see **Appendix** A.2.1 for detailed analysis), but they are designed without full consideration for the nature of classification and how humans do it. For a specific scenario, the existing classifiers often perform well on one evaluation dimension but poorly on others. For example, deep neural network has exceed the human level in very specific settings [8]. Unfortunately, its working mechanism is difficult to be understood due to the complexity of the network structure [9, 10], and it is also very vulnerable to be attacked [11, 12]. Instead, human can perform well on a variety of tasks and on different evaluation dimensions.

To implement a classifier that is easy to understand and interpretable, one effective approach is to draw on relevant research in cognitive science. As mentioned above, classifying is essentially the

*Corresponding author

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

process of concept cognition. In cognitive science, there is a lot of valuable theories about how human learn, represent and use concept [13]. However, few of the existing classifiers are designed based on these theories, which leads to that the existing classifiers are unable to take into account the performance on multiple evaluation dimensions. Inspires by this, we are encouraged to design a new classifier based on concept cognition, which can simultaneously perform well on different evaluation dimensions. In this paper, a new learning machine is designed based on the following principles.

Similarity is the basis of concept representation. Similarity is a fundamental concept within cognitive science and plays a crucial role in the process of human classification [14]. Concepts are represented based on similarity for children, which is also a basic choice for adults [15]. Based on this principle, we prove the equivalence between classification and equivalence relation (ER) via binary relation in set theory. Considering that ER is a special similarity, we actually demonstrate that all classification problems can be solved based on similarity (see **Section** 2.1).

Concept is fuzzy. An important conclusion in cognitive science is that fuzziness is the intrinsic property of concept [13]. The fuzziness and the randomness are the two most important sources of the uncertainty. In machine learning, probability theory is often used for dealing with the uncertainty, such as Bayesian methods [16, 17], label distribution learning [18], etc. It is sufficient for probability theory to deal with the randomness rather than the fuzziness [19]. However, the intrinsic property of concept is just the fuzziness rather than the randomness. An intuitive example is shown in Figure 1.

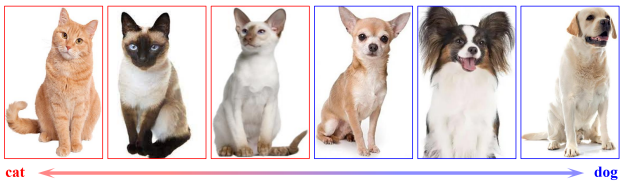

Figure 1: The visual concepts of 'cat' and 'dog' are of fuzziness. In fuzzy set theory, the membership degree of the 3rd image to 'cat' ('dog') can be 0.6 (0.5). In probability theory, the probability of the 3rd image being 'cat' ('dog') can be 0.6 ($1 - 0.6 = 0.4$). Obviously, the latter is unreasonable, because the 3rd image can not be 'cat' in some case and be 'dog' in the other case.

Based on this principle, fuzzy equivalence relation (FER) in fuzzy set theory [20] rather than ER is chosen as the core component of the new learning machine (see **Section** 2.2).

Exemplar theory based concept representation In cognitive science, there are several theories of concept representation [13]. Exemplar theory is chosen to solve concept representation problem, in this paper. Because it not only has a good self-consistency in cognitive science, but also it can be naturally integrated with the data-driven machine learning paradigm (see **Section** 2.3).

To summarize, the main contributions of this paper are four-folds as below:
(1) This paper revisits classification problem from the perspective of concept cognition, and proves the equivalence between classification problem and ER. Furthermore, considering that concept is intrinsically fuzzy, the classification problem is modeled as an FER problem.
(2) A new and general learning machine FLM is proposed based on concept cognition for solving FER problem.
(3) A new fuzziness permissible loss is designed. Given the new loss, this paper prove that FER can be approximated effectively by fuzzy similarity relation (FSR). And then a neural network based FLM (NN-FLM) is designed.
(4) Extensive experimental results demonstrate the superiority of NN-FLM on different evaluation dimensions including interpretability, robustness, and generalization performance.

2 Fuzzy Learning Machine

Basic notations Let $\mathcal{X}$ and $\mathcal{Y} = \{c_1, c_2, \cdots, c_k\}$ be the input and output spaces, respectively. Let $\varphi : \mathcal{X} \to \mathcal{Y}$ be the unknown target function and $\forall x \in \mathcal{X}$, $\varphi(x) \in \mathcal{Y}$ is the true class label of the sample x. The goal of classification is to approximate φ as closely as possible. Let $D_{\text{train}} = \{(x_i, y_i) | x_i \in \mathcal{X}, y_i = \varphi(x_i) \in \mathcal{Y}, i = 1, 2, \cdots, n\}$ and $D_{\text{test}} = \{\ x_j |\ x_j \in \mathcal{X},\ j = n+1, n+2, \cdots, n+m\ \}$ be the training and test data sets, respectively.

2.1 Classification Problem Can Be Solved Based on Similarity

Researches [14, 15] on cognitive science prove that similarity is the basis of concept representation. However, the relationship between similarity and classification (e.g. whether all classification problems can be solved by using similarity) still lacks a mathematical justification. This section will accomplish it. The mathematical tool used in this section is binary relation in set theory. See **Appendix** A.1.2 for a brief introduction.

First, the formal definition of classification problem is given.

Definition 1 *For an input space $\mathcal{X}$, an output space $\mathcal{Y} = \{c_1, c_2, \cdots, c_k\}$, and an unknown target function $\varphi : \mathcal{X} \to \mathcal{Y}$. The $(\mathcal{X}, \mathcal{Y}, \varphi)$-classification problem is defined as: $\forall x \in \mathcal{X}$, finding $\varphi(x)$.*

It should be pointed out that the **Definition** 1 is so general that almost all classification problems studied in machine learning can be represented by it.

Definition 2 *Given a $(\mathcal{X}, \mathcal{Y}, \varphi)$-classification problem, (1) let $\mathcal{X} \times \mathcal{X}$ be the new input space, (2) let $\{0, 1\}$ be the new output space, (3) let $\varphi^\dagger : \mathcal{X} \times \mathcal{X} \to \{0, 1\}$ be the new target function, where $\forall (x_i, x_j) \in \mathcal{X} \times \mathcal{X}, \varphi^\dagger ((x_i, x_j)) = \mathbb{I} (\varphi (x_i) = \varphi (x_j))$, then finding the $\varphi^\dagger$ is defined as its adjoint $\left(\mathcal{X} \times \mathcal{X}, \{0, 1\}, \varphi^\dagger\right)$-classification problem.*
*Obviously, the $\varphi^\dagger$ is an equivalence relation (ER) on $\mathcal{X}$ (see **Definition** 7 in **Appendix** A.1.2) and the adjoint $\left(\mathcal{X} \times \mathcal{X}, \{0, 1\}, \varphi^\dagger\right)$-classification problem is a binary-classification problem.*

Based on the above definitions, we have the following proposition.

Proposition 1 *The $(\mathcal{X}, \mathcal{Y}, \varphi)$-classification problem is equivalent to its adjoint $\left(\mathcal{X} \times \mathcal{X}, \{0, 1\}, \varphi^\dagger\right)$-classification problem, i.e. if one problem is solved, then the other problem will also be solved. (See **Appendix** A.3.1 for proof)*

The **Proposition** 1 establishes the relationship between ER and classification problem. For any $(\mathcal{X}, \mathcal{Y}, \varphi)$-classification problem, we can always solve it equivalently by solving its adjoint $\left(\mathcal{X} \times \mathcal{X}, \{0, 1\}, \varphi^\dagger\right)$-classification problem. Because ER $\varphi^\dagger : \mathcal{X} \times \mathcal{X} \to \{0, 1\}$ is a special similarity on $\mathcal{X}$. Therefore, we can solve any classification problem by finding a appropriate similarity.

2.2 Capture Fuzziness of Concept

The mathematical tools used in this section are fuzzy set theory and binary fuzzy relation. See **Appendix** A.1.3 for a brief introduction.

A large number of theories and experiments in cognitive science have shown that fuzziness is an intrinsic property of concepts [13]. A short mathematical argument is as follows. For any $(\mathcal{X}, \mathcal{Y}, \varphi)$-classification problem, we have $|\mathcal{X}| > |\mathcal{Y}|$. So $\exists x_i, x_j \in \mathcal{X}, x_i \neq x_j$, such that $\varphi (x_i) = \varphi (x_j)$. The membership degrees of x_i and x_j to the concept $\varphi (x_i)$ are equal even if x_i and x_j may be very different. When fuzziness of concept is introduced, the dilemma can be solved directly. Because x_i and x_j are different, their membership degrees to the concept $\varphi (x_i)$ are different. As shown in Figure 1, the membership degree of the 1st image to 'cat' is greater than the 3rd image, and the membership degree of 3rd image to 'dog' is less than the 6th image.

The **Section** 2.1 concludes that any classification problem can be solved based on the special similarity, i.e. $\forall x_i, x_j \in \mathcal{X}$, either they belong to the same concept then $\varphi^\dagger ((x_i, x_j)) = 1$, or they don't belong to the same concept then $\varphi^\dagger ((x_i, x_j)) = 0$. Obviously, this assumption is hard to satisfy in real world because it does not capture the intrinsic fuzziness of concept.

Based on the above analysis, a new mathematical tool is needed to effectively deal with the fuzziness of concept. Fortunately, an effective solution is found in the fuzzy set theory that is introduced by Lotfi A. Zadeh [20] in 1965. In it, the crisp definition of the set is extended to a fuzzy one. Specifically, in set theory, the membership degree of an object to a set is crisp, i.e. either it belongs to the set then membership degree is 1, or it does not belong to the set then the membership degree is 0. Instead, in fuzzy set theory, the membership degree of an object to a set is fuzzy, which is represented by a real number in the interval $[0, 1]$ (see **Definition** 11 in **Appendix** A.1.3 and Figure 1).

To capture the fuzziness of concept, ER in **Definition** 2 is replaced by FER (see **Definition** 16 and **Theorem** 3 in **Appendix** A.1.3), which leads to the following definition.

Definition 3 *Given a $(\mathcal{X}, \mathcal{Y}, \varphi)$-classification problem, (1) let $\mathcal{X} \times \mathcal{X}$ be the new input space, (2) let $[0,1]$ be the new output space, (3) let $\varphi^{\ddagger} : \mathcal{X} \times \mathcal{X} \to [0,1]$ be the new target function, where $\varphi^{\ddagger}$ is an FER on $\mathcal{X}$, and $\forall (x_i, x_j) \in \mathcal{X} \times \mathcal{X}$, $\varphi^{\ddagger}((x_i, x_j))$ be the degree that x_i and x_j belong to the same concept. Then finding the $\varphi^{\ddagger}$ is defined as its adjoint $\left(\mathcal{X} \times \mathcal{X}, [0,1], \varphi^{\ddagger}\right)$-FER problem.*

To solve the $(\mathcal{X}, \mathcal{Y}, \varphi)$-classification problem, it is more reasonable to solve the $\left(\mathcal{X} \times \mathcal{X}, [0,1], \varphi^{\ddagger}\right)$-FER problem than the $\left(\mathcal{X} \times \mathcal{X}, \{0,1\}, \varphi^{\dagger}\right)$-classification problem. Because it is fuzzy whether two samples belong to the same concept in the former. This assumption fits well with the principle that concept is fuzzy. For a learning task, solving the $\left(\mathcal{X} \times \mathcal{X}, [0,1], \varphi^{\ddagger}\right)$-FER problem is finding an FER on $\mathcal{X}$ that can approximate φ by using D_{train}. The process can be formally described as follows

$$\mathrm{f}^* = \arg\min_{\mathrm{f}} \mathcal{L}\left(D_{\text{train}}, \mathrm{f}\right) + \gamma \mathcal{R}(\mathrm{f}) \ s.t. \ \mathrm{f} \in \{\mathrm{g} | \mathrm{g} \text{ is an FER on } \mathcal{X}\}, \tag{1}$$

where $\mathcal{L}$ is the loss function that measures how well the f fits the training data set D_{train}, $\mathcal{R}$ is the regularization term, and $\gamma > 0$ is the tradeoff parameter.

2.3 Exemplar Theory Based Concept Representation

The main knowledge used in this section is concept representation theory. See **Appendix** A.1.1 for a brief introduction.

Suppose we have obtained an FER f^* (the optimal solution of formula (1)) by using D_{train}. From the perspective of cognitive science, we have constructed the basis used for representing all concepts in $\mathcal{Y}$. Next, we need to complete concept representation based on the basis.

In cognitive science, exemplar theory is about concept representation [13]. Its core idea is that human represents a concept by remembering some objects that belong to it. Therefore, we need to find some exemplars that can represent every concept in $\mathcal{Y}$. In cognitive science, which object is suitable as an exemplar of a concept is still an open question, but it is an unequivocal fact that the similarity is the key of exemplar theory [13]. In a data-driven learning task, the exemplar set of every concept can be determined by the following definition.

Definition 4 *Given a $(\mathcal{X}, \mathcal{Y}, \varphi)$-classification problem and a training data set D_{train}. Let f^* be the FER found by using D_{train}. $\forall c \in \mathcal{Y}$, let $X^c_{\text{train}} = \{x_i | (x_i, y_i) \in D_{\text{train}}, y_i = c\}$. And $\forall x \in X^c_{\text{train}}$, let $\mu(x, c) = \frac{1}{|X^c_{\text{train}}|} \sum_{x_i \in X^c_{\text{train}}} \mathrm{f}^*((x, x_i))$. Then, $\forall c \in \mathcal{Y}$, the exemplar set is defined as*

$$E_c = \{x \,|x \in X^c_{\text{train}}, \mu(x, c) \text{ is the top}-n^c_{\text{exe}} \text{ largest value in } \{\mu(x_i, c) \,|x_i \in X^c_{\text{train}}\}\},$$

where n^c_{exe} is a manually specified parameter.

The above definition is intuitive. Take Figure 1 as an example. Among the 3 images of 'cat', the 1st image is more suitable as an exemplar of concept 'cat' than the other 2 images, because the 1st image is more similar to the 3 images of 'cat' than the other 2 images.

After obtaining the exemplar set of every concept, the predicting process can be described as

$$\hat{y} = \arg\max_{c \in \mathcal{Y}} \frac{1}{|E_c|} \sum_{x_i \in E_c} \mathrm{f}^*((x, x_i)), \ \forall x \in \mathcal{X}. \tag{2}$$

2.4 The Framework of FLM

Combining the **Sections** 2.1-2.3, the training and test processes of FLM are given as follows.

Algorithm 1: The training process of FLM

Input: The training data set D_{train}, the number of exemplars of every class $n^c_{\text{exe}}, \forall c \in \mathcal{Y}$.
Find f^* by solving optimization problem (1).
for $c \in \mathcal{Y}$ **do**
  Compute the exemplar set E_c of class c by **Definition** 4.
end
Output: f^* and $E_c, \forall c \in \mathcal{Y}$.

Algorithm 2: The test process of FLM

Input: The test data set D_{test}, the learned f^*, and the exemplar set $E_c, \forall c \in \mathcal{Y}$.
for $x_j \in D_{\text{test}}$ **do**
  Compute the predicting label $\hat{y}_j$ of sample x_j by formula (2).
end
Output: $(x_j, \hat{y}_j), \forall x_j \in D_{\text{test}}$.

2.5 Relationship to the Existing Methods

The existing classifiers [1] aim to learn a function $\mathrm{f}: \mathcal{X} \to \mathcal{Y}$, while FLM aims to learn an FER $\mathrm{f}_{\mathrm{FER}}: \mathcal{X} \times \mathcal{X} \to [0,1]$. Therefore, the existing classifiers and FLM solve classification problem from different perspectives. FLM is designed based on concept cognition, while most existing classifiers seldom fully consider concept cognition. There is also a large corpus fuzzy classifiers [21] that use fuzzy set theory for classification task. Compared with most existing fuzzy classifiers, FLM is more suitable for data-driven learning tasks.

The distance metric learning [22, 23, 24], the siamese network [25, 26, 27], and the relation network based methods [28, 29, 30] both aim to learn a similarity $\mathrm{s}: \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. And FLM aims to learn an FER $\mathrm{f}_{\mathrm{FER}}: \mathcal{X} \times \mathcal{X} \to [0,1]$. On the one hand, the $\mathrm{f}_{\mathrm{FER}}$ is a special similarity. So these three methods and FLM are similar in this respect. On the other hand, the $\mathrm{f}_{\mathrm{FER}}$ needs to satisfy four properties (i.e. normalization, reflexivity, symmetry, and transitivity), while the s usually cannot satisfy these properties at the same time. Therefore these methods can not fully capture the nature of classification.

The detailed analysis about the above conclusions can be found in **Appendix** A.2.

3 NN-FLM: A Neural Network Based FLM on $\mathbb{R}^d$

In this section, the most popular input space, $\mathcal{X} = \mathbb{R}^d$, is taken into account. And the goal is to instantiate optimization problem (1) and solve it. The mathematical tools used in this section are binary fuzzy relation and the transitive-closure. See **Appendix** A.1.3 for a brief introduction. And the code implementation is provided in supplementary material.

3.1 The Separability of Adjoint $\left(\mathbb{R}^d \times \mathbb{R}^d, \{0,1\}, \varphi^\dagger\right)$-Classification Problem

To solve the adjoint $\left(\mathbb{R}^d \times \mathbb{R}^d, \{0,1\}, \varphi^\dagger\right)$-classification problem described in the **Definition** 2, classifiers need to deal with pairs of samples on $\mathbb{R}^d$. One of the simplest methods is to concatenate the feature vectors of a pair of samples, i.e.

$$D_{\mathrm{train}}^{\mathrm{concat}} = \left\{\left([\mathbf{x}_i; \mathbf{x}_j], \mathbb{I}\left(y_i = y_j\right)\right) \mid \left(\mathbf{x}_i, y_i\right), \left(\mathbf{x}_j, y_j\right) \in D_{\mathrm{train}}\right\}, \tag{3}$$

where $\forall \mathbf{a}, \mathbf{b} \in \mathbb{R}^d$, $[\mathbf{a}; \mathbf{b}] \in \mathbb{R}^{2d}$ denotes the concatenation of them. Then, a pair of samples can be converted into a composite sample, which can be processed by classifiers designed for $\mathbb{R}^{2d}$. However, the binary-classification problem obtained by the above method is not linearly separable. See the following theorem.

Theorem 1 *For a $\left(\mathbb{R}^d, \mathcal{Y}, \varphi\right)$-classification problem and a training data set D_{train}, if the sample is defined as formula* (3)*, then the corresponding $\left(\mathbb{R}^{2d}, \{0,1\}, \varphi^\dagger\right)$-classification problem is not linearly separable. (see **Appendix** A.3.2 for proof)*

Therefore, FLM on $\mathbb{R}^d$ must be able to handle the linear inseparability of the training set.

3.2 The Design of NN-FLM

To solve the original $\left(\mathbb{R}^d, \mathcal{Y}, \varphi\right)$-classification problem, an FER on $\mathbb{R}^d$ (see **Definition** 3) should be found. It is very difficult to directly obtain a binary fuzzy relation that satisfies transitivity. So an indirect method is adopted. First, construct a binary fuzzy relation that satisfies reflexivity and symmetry, i.e. FSR (see **Definition** 15 in **Appendix** A.1.3). Then convert the FSR into FER.

To fully implement FLM, the following requirements must be considered. (a) The linear inseparability of the problem (see **Theorem** 1). (b) The learned function must be an FER, i.e. the learned function is normalized, reflexive, symmetric, and transitive. (c) The loss function must consider the fuzziness of concept. (d) An effective optimizer must be designed, and it should be efficient especially for large scale training data sets.

Based on the above analysis, NN-FLM includes three components: (1) Fuzzy similarity relation network, which responds to (a) and the first 3 conditions in (b), (2) Fuzziness permissible loss, which responds to (c) and indirectly satisfies the transitivity in (b), and (3) Stochastic gradient descent based optimizer, which responds to (d). An overall design of NN-FLM is shown in Figure 2.

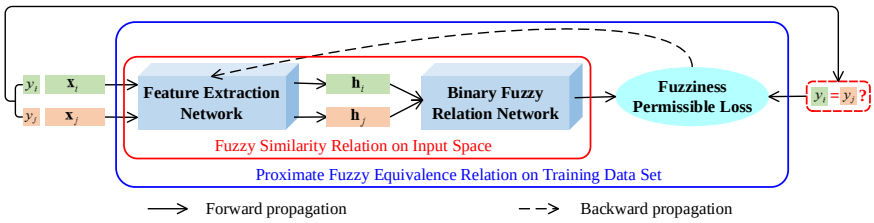

Figure 2: The overall design of NN-FLM.

(1) Fuzzy similarity relation network contains two parts: feature extraction network and binary fuzzy relation network.

First, the feature extraction network can be formally described as follows

$$\forall \mathbf{x} \in \mathbb{R}^d, \mathrm{h}(\mathbf{x}; \Theta) \in \mathbb{R}_+^{d_h}, \tag{4}$$

where $\mathbb{R}_+$ is the nonnegative real numbers, d_h is the dimension of the latent space, and Θ is the set of learnable parameters. A multi-layer neural network with nonlinear activation function is recommended, to deal with the linear inseparability of the problem.

Second, the cosine similarity is used as the skeleton of the binary fuzzy relation network, i.e. $\forall \mathbf{h}_i, \mathbf{h}_j \in \mathbb{R}_+^{d_h}, \mathrm{g}(\mathbf{h}_i, \mathbf{h}_j) = \frac{\mathbf{h}_i^T \mathbf{h}_j}{\|\mathbf{h}_i\|_2 \|\mathbf{h}_j\|_2}$. Let $\mathrm{f}((\mathbf{x}_i, \mathbf{x}_j); \Theta) = \mathrm{g}(\mathrm{h}(\mathbf{x}_i; \Theta), \mathrm{h}(\mathbf{x}_j; \Theta)), \forall \mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^d$, be the composite of h and g. Obviously, f is an FSR on $\mathbb{R}^d$, i.e. f is normalized, reflexive, and symmetric.

It is very difficult to make f satisfy transitivity on $\mathbb{R}^d$. However, it is easy to make f satisfy transitivity on the training sample set $X_{\text{train}} = \{\mathbf{x}_i | (\mathbf{x}_i, y_i) \in D_{\text{train}}\}$. Let $\mathbf{S} \in [0,1]^{n \times n}, s_{ij} = \mathrm{f}((\mathbf{x}_i, \mathbf{x}_j)), \forall i, j = 1, 2, \cdots, n$, be the FSR matrix on X_{train} (see **Definition** 14 in **Appendix** A.1.3). And let

$$\mathbf{T} = \mathrm{t}_{\text{FSR2FER}}(\mathbf{S}) \tag{5}$$

be the transitive-closure of $\mathbf{S}$ (see **Definition** 20, **Theorem** 4 and 5 in **Appendix** A.1.3). Then, $\mathbf{T} \in [0,1]^{n \times n}$ is an FER matrix on X_{train}.

(2) Fuzziness permissible loss For a pair of training samples $(\mathbf{x}_i, y_i), (\mathbf{x}_j, y_j) \in D_{\text{train}}$, let t_{ij} be the predicting FER value of $(\mathbf{x}_i, \mathbf{x}_j)$ by formula (5). The (α, β)-insensitive loss is used to measure the difference between the predicting value and the true value and can be written as

$$\mathcal{L}_{\alpha,\beta}(t_{ij}, y_i, y_j) = \begin{cases} \max\{t_{ij} - \alpha, 0\}, & y_i \neq y_j \\ \max\{\beta - t_{ij}, 0\}, & y_i = y_j \end{cases}, \tag{6}$$

where $0 \leq \alpha < 0.5 < \beta \leq 1$ are two hyper-parameters. The gap $\beta - \alpha \in (0, 1]$ is used for controlling the fuzziness degree of concept. And the greater the value is, the less the fuzziness will be. Therefore, the loss on the entire training data set is

$$\mathcal{L}_{\text{FER}} = \sum_{i=1}^{n} \sum_{j=1}^{n} \mathcal{L}_{\alpha,\beta}(t_{ij}, y_i, y_j). \tag{7}$$

However, the time complexity of converting the FSR matrix $\mathbf{S}$ into the FER matrix $\mathbf{T}$ is $O(n^3 \log_2 n)$ (see **Remark** in **Appendix** A.1.3). In practice, it is not feasible to directly optimize the above loss, especially when the number of training samples n is large. Fortunately, due to the intrinsic property of the transitive-closure, we can indirectly control the loss (7) by optimizing the following loss

$$\mathcal{L}_{\text{FSR}} = \sum_{i=1}^{n} \sum_{j=1}^{n} \mathcal{L}_{\alpha,\beta}(s_{ij}, y_i, y_j). \tag{8}$$

See the following theorem.

Theorem 2 *Given a training set* $D_{\text{train}} = \{(\boldsymbol{x}_i, y_i) | \boldsymbol{x}_i \in \mathbb{R}^d, y_i = \varphi(\boldsymbol{x}_i), i = 1, 2, \cdots, n\}$ *and an FSR* f *on* $X_{\text{train}} = \{\boldsymbol{x}_i | (\boldsymbol{x}_i, y_i) \in D_{\text{train}}\}$. *Let* $\boldsymbol{S} \in [0,1]^{n \times n}$, $s_{ij} = \mathrm{f}((\boldsymbol{x}_i, \boldsymbol{x}_j))$, $\forall i, j = 1, 2, \cdots, n$, *be the FSR matrix on* X_{train}, *and* $\boldsymbol{T} = \mathrm{t}_{\text{FSR2FER}}(\boldsymbol{S})$ *be the transitive-closure of* $\boldsymbol{S}$. $\forall\ 0 \leq \alpha < 0.5 < \beta \leq 1$, *if* $\sum_{i=1}^{n} \sum_{j=1}^{n} \mathcal{L}_{\alpha,\beta}(s_{ij}, y_i, y_j) = 0$, *then* $\sum_{i=1}^{n} \sum_{j=1}^{n} \mathcal{L}_{\alpha,\beta}(t_{ij}, y_i, y_j) = 0$. *(see* ***Appendix*** *A.3.3 for proof)*

According to the **Theorem** 2, when the loss (8) reaches its minimum 0, the loss (7) also reaches its minimum 0. The loss (8) can be optimized without calculating the FER matrix on all training samples. Therefore, the loss (8) is used to complete the learning process, in this paper.

(3) Stochastic gradient descent based optimizer To sum up, the learning process of NN-FLM can be written as

$$\min_{\Theta} \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \mathcal{L}_{\alpha,\beta} \left(\mathrm{f} \left(\left(\mathbf{x}_i, \mathbf{x}_j \right) ; \Theta \right), y_i, y_j \right) + \gamma \mathcal{R}(\Theta). \tag{9}$$

The time complexity of calculating the entire FSR matrix $\mathbf{S}$ is at least $O\left(n^2\right)$, which still has a large computational burden. Fortunately, to optimize the above loss, it is not necessary to calculate the entire matrix $\mathbf{S}$ at the same time. So the above optimization problem can be solved efficiently using the stochastic gradient descent with mini-batch, even when n is large.

4 Experiments

4.1 An Example on MNIST Data Set

Experimental settings In this section, the MNIST data set [31] is chosen to demonstrate the working mechanism of NN-FLM. In NN-FLM, a 5-layer convolutional neural network is used as the feature extraction network, and the fuzzy parameters are fixed as $\alpha = 0.2$, $\beta = 0.8$. See **Appendix** A.4.1 for more experimental details.

Interpretability analysis First, Figure 3a shows the FSR matrix predicted by NN-FLM on 10,000 test samples. The following observations can be seen from it. (1) The FSR value between samples from the same classes is significantly larger than that of samples from different classes. (2) The FSR matrix exhibits an obvious block-diagonal structure, where each block corresponds to a class. Combining (1) and (2), we can conclude that NN-FLM can learn a high-quality FSR from the training data. Based on the learned FSR, the concepts of 10 visual digits can be accurately represented. Second, Figure 3b shows the exemplars selected from the training set according to **Definition** 4. The following observations can be seen from it. (1) The 5 exemplars of every concept are regular, with no interrupted strokes, no distortions, well-proportioned parts of the digits, and the digits are located in the center of the image. It shows that the exemplars of the every concept selected by NN-FLM can accurately capture the characteristics of every concept. (2) There are certain changes among the 5 exemplars of every concept, which ensures that the exemplars can cover the intra-concept variations.

In summary, we can conclude that NN-FLM can learn high-quality FSR from training data. And then based on learned FSR, the high-quality exemplars can be selected to accurately represent the corresponding concept.

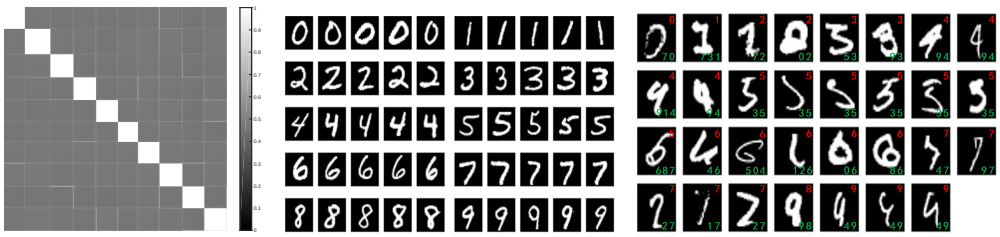

Figure 3: (a) The predicting FSR matrix on the test set (the samples are sorted by class labels). (b) The exemplars of 10 classes selected from the training set. (c) The 31 errors of NN-FLM, with annotated label (up right) and top-2 or top-3 predicting labels (down right).

Robustness analysis A total of 31 test samples are misclassified and they are shown in Figure 3c. The following conclusions can be drawn from it. (1) All the misclassified samples are irregular digital images. (2) Among them, 26 images (3 images) are hit by the 2nd (3rd) class label predicted by NN-FLM. (3) For the 2 images (1st and 3rd in row 3 of Figure 3c), none of the top-3 class labels predicted by NN-FLM hits the annotated label. These 31 images are so irregular that it is difficult

to give crisp and uncontroversial labels even for humans. In this case, the class labels predicted by NN-FLM are in line with human cognition to a certain degree.

In summary, we can conclude that NN-FLM can adequately capture fuzziness of concept. Therefore, the samples with a high degree of fuzziness can be identified and assigned reasonable candidate concepts by NN-FLM. This verifies that NN-FLM is robust to the controversial samples.

4.2 Comparison With 179 Classifiers on 121 Data Sets

Experimental settings In the literature [1], 179 classifiers from 17 families (see Table 3 in **Appendix** A.4.2) are systematically compared on 121 benchmark data sets (see Table 2 in **Appendix** A.4.2). In order to verify the generalization performance of FLM, NN-FLM is chosen and compared with the above 179 classifiers. In all experiments, NN-FLM adopts a 3-layer fully connected network as the feature extraction network. The fuzzy parameters are fixed as $\alpha = 0.2, \beta = 0.8$. More details about the experimental settings can been found in **Appendix** A.4.2.

Table 1: The rank of NN-FLM among 180 classifiers on 121 data sets

ID	Rank	ID	Rank	ID	Rank	ID	Rank	ID	Rank	ID	Rank
1	1	21	1	41	1	61	2	81	10	101	21
2*	1	22	1	42	1	62*	2	82	10.5	102	23
3	1	23	1	43	1	63	2.5	83	11	103*	23.5
4	1	24	1	44	1.5	64*	2.5	84	12	104	24.5
5	1	25	1	45	1.5	65	3	85	13	105	26
6	1	26	1	46	1.5	66	3	86	13.5	106	26
7	1	27	1	47	1.5	67	3	87	14	107	29
8	1	28	1	48*	1.5	68	3	88*	14.5	108	31.5
9	1	29	1	49*	1.5	69	3.5	89	15	109	32
10	1	30	1	50	1.5	70*	4	90	15	110	35
11	1	31	1	51	1.5	71	4.5	91*	15	111	36
12	1	32	1	52	1.5	72	5	92	15.5	112	38
13	1	33	1	53	1.5	73	5.5	93	15.5	113	39
14	1	34	1	54*	2	74	6	94	16	114	39
15	1	35	1	55	2	75	6.5	95	16	115	39
16	1	36*	1	56	2	76	7.5	96	16.5	116	42
17	1	37	1	57	2	77	8	97	17.5	117	47
18	1	38	1	58	2	78	9	98	18.5	118	60
19	1	39	1	59*	2	79	9	99	19	119	62
20	1	40	1	60	2	80	10	100	20	120	63.5
										121	65

(a) ID: The unique ID of every data set (see Table 2 in **Appendix** A.4.2). (b) $*$: The 12 small sample data sets (see Table 5 in **Appendix** A.4.2). (c) The rank considers that two or more classifiers have the same test accuracy. For example, on data set 121 there are 129 classifiers with 100% test accuracy, so the ranks of these 129 classifiers are $\left(\sum_{r=1}^{129} r\right)/129 = 65$.

The top-10 classifiers The 10 classifiers with the smallest mean rank among 180 classifiers on 121 data sets are selected for comparison analysis. They are NN-FLM (1st), 3 classifiers from the random forest family (2nd parRF-t, 3rd rf-t, 5th rforest-R), 3 classifiers from the support vector machine family (4th svm-C, 6th svmPoly-t, 8th svmRadCost-t), 1 classifier from the decision tree family (7th C5.0-t), and 2 classifiers from the neural network family (9th elm-kernel-m, 10th avNNet-t).

Accuracy analysis The test accuracy of the 179 comparison methods are from the literature [1]. We run NN-FLM on 121 data sets and record the test accuracy. Then the accuracy of 180 classifiers are used for evaluating their performance. The following conclusions can be drawn from it. (1) Due to space constraint, the accuracy of NN-FLM and other top-9 comparison methods are shown in Table 4 in **Appendix** A.4.2. And it can be seen that the mean accuracy on 121 data sets of NN-FLM is 5.49% higher than the 2nd parRF-t and 6.00% higher than the 3rd rf-t. Therefore, NN-FLM achieves a significant improvement compared with the 179 classifiers. (2) The rank of NN-FLM among 180

methods on 121 data sets is recorded in Table 1. And it can be seen that NN-FLM ranks first on 43 data sets (data sets 1-43) and ties for first place on 10 data sets (data sets 44-53), and ranks in the top-3 on more than half of the data sets (data sets 1-68). Therefore, NN-FLM achieves outstanding rank among the the 180 classifiers.

Comparative advantage analysis Figure 4a shows the proportion of the top-10 classifiers ranked in top-K ($K = 1, 3, 5, 10, 15, 20$) among the 121 data sets. And Figure 4b shows the mean and standard deviation of the rank of the top-10 classifiers on 121 data sets. The following conclusions can be drawn from it. (1) NN-FLM ranks the top-1 on $43/121 \approx 35.5\%$ of the data sets and the top-3 on $68/121 \approx 56.2\%$ of the data sets. And NN-FLM ranks in top-20 on more that 80% of the data sets. (2) None of the other 9 comparison classifiers ranks in top-20 on more than 50% of the data sets. (3) NN-FLM has the smallest mean rank 10.51. And the mean rank is significantly superior to the second-ranked classifier, parFR-t, 33.50. (4) NN-FLM has the smallest standard deviation 14.86. And the standard deviation is significantly lower than to the other comparison classifiers. In summary, the performance of NN-FLM is significantly superior to the comparison classifiers. At the same time, it can perform well on various tasks. In contrast, the comparison classifiers can only perform well on limited tasks. Therefore, the performance of NN-FLM is more stable than the others. The possible reason is that NN-FLM can adequately capture the nature of classification.

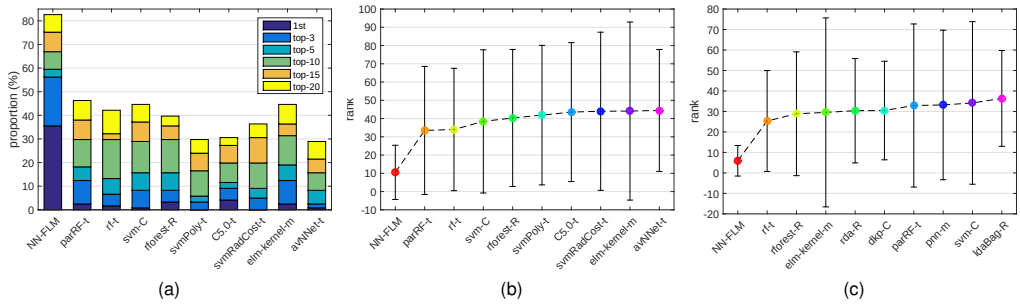

Figure 4: The statistics on top-10 methods among 180 classifiers. (a) The proportion of top-K rank on 121 data sets. (b) The mean and standard deviation of the rank on 121 data sets. (c) The mean and standard deviation of the rank on 12 small data sets.

Experiments on small sample data sets To verify the performance of NN-FLM on small sample classification problem, we selected 12 data sets with small number of samples (see Table 5 in **Appendix** A.4.2) from the 121 data sets. In these data sets, the mean number of training samples for each class is less than 15.

We selected the top-10 methods with the smallest average rank on the 12 small sample data sets for comparison, see Figure 4c. The following conclusions can be drawn from it. (1) Compared Figures 4b with 4c, it can be seen that the top-10 classifiers on the 12 small sample data sets differ from that on the all 121 data sets. Some classifiers (svmPoly-t, C5.0-t, svmRadCost-t, and avNNet-t) rank in the top-10 on the all data sets but not on the small sample data sets. (2) The mean rank of NN-FLM on the 12 small sample data sets is still the best. (3) The rank of NN-FLM is significantly superior to the second-ranked method, fr-t (5.91 ± 7.44 vs 25.33 ± 24.62). In summary, it can be concluded that when there are fewer training samples, the superiority of NN-FLM is more significant. The reason is that NN-FLM can capture the characteristic and intra-concept variations of concept through a small number of exemplars. Therefore, the exemplar theory based concept representation is reasonable.

5 Conclusion

A new learning machine based on concept cognition, fuzzy learning machine, is proposed. The new learning machine has the advantages of strong interpretability and solid mathematical foundation. And it is of certain degree of robustness because it can effectively deal with the intrinsic fuzziness in classification. At the same time, a large number of comparison experiments show that it can reach an advanced level even with a simple implementation.

Acknowledgments and Disclosure of Funding

This work is supported by the National Key Research and Development Program of China (under grant 2020AAA0106100).

References

[1] Manuel Fernández Delgado, Eva Cernadas, Senén Barro, and Dinani Gomes Amorim. Do we need hundreds of classifiers to solve real world classification problems? *Journal of Machine Learning Research*, 15(1):3133–3181, 2014.

[2] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

[3] Zhifeng Kong and Kamalika Chaudhuri. Understanding instance-based interpretability of variational auto-encoders. In *Advances in Neural Information Processing Systems, virtual*, 2021.

[4] Aya Abdelsalam Ismail, Héctor Corrada Bravo, and Soheil Feizi. Improving deep learning interpretability by saliency guided training. In *Advances in Neural Information Processing Systems, virtual*, 2021.

[5] Cheng Zhang, Kun Zhang, and Yingzhen Li. A causal view on robustness of neural networks. In *Advances in Neural Information Processing Systems, virtual*, pages 289–301, 2020.

[6] Shahd Safarani, Arne Nix, Konstantin Willeke, Santiago A. Cadena, Kelli Restivo, George H. Denfield, Andreas S. Tolias, and Fabian H. Sinz. Towards robust vision by multi-task learning on monkey visual cortex. In *Advances in Neural Information Processing Systems, virtual*, 2021.

[7] Songwei Ge, Vasu Singla, Ronen Basri, and David W. Jacobs. Shift invariance can reduce adversarial robustness. In *Advances in Neural Information Processing Systems, virtual*, 2021.

[8] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[9] Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA*, pages 4829–4837, 2016.

[10] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA*, pages 5188–5196, 2015.

[11] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations, Banff, AB, Canada*, 2014.

[12] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations, Vancouver, BC, Canada*, 2018.

[13] Gregory Murphy. *The Big Book of Concepts*. MIT press, 2004.

[14] Ulrike Hahn. Similarity. *Wiley Interdisciplinary Reviews: Cognitive Science*, 5(3):271–280, 2014.

[15] Heidi Kloos and Vladimir M Sloutsky. What's behind different kinds of kinds: effects of statistical density on learning and representation of categories. *Journal of Experimental Psychology: General*, 137(1):52–72, 2008.

[16] Pedro M. Domingos and Michael J. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29(2-3):103–130, 1997.

[17] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Series in Representation and Reasoning. Morgan Kaufmann, 1989.

[18] Xin Geng. Label distribution learning. *IEEE Transactions on Knowledge and Data Engineering*, 28(7):1734–1748, 2016.

[19] Lotfi A. Zadeh. Is probability theory sufficient for dealing with uncertainty in AI: A negative view. In *Annual Conference on Uncertainty in Artificial Intelligence, Los Angeles, CA, USA*, pages 103–116, 1985.

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

[21] Ludmila I. Kuncheva. *Fuzzy Classifier Design*, volume 49 of *Studies in Fuzziness and Soft Computing*. Springer, 2000.

[22] Eric P. Xing, Andrew Y. Ng, Michael I. Jordan, and Stuart J. Russell. Distance metric learning with application to clustering with side-information. In *Advances in Neural Information Processing Systems, Vancouver, British Columbia, Canada*, pages 505–512, 2002.

[23] Dewei Li and Yingjie Tian. Survey and experimental study on metric learning methods. *Neural Networks*, 105:447–462, 2018.

[24] Mang Ye, Jianbing Shen, Gaojie Lin, Tao Xiang, Ling Shao, and Steven C.H. Hoi. Deep learning for person re-identification: A survey and outlook. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021, DOI: 10.1109/TPAMI.2021.3054775.

[25] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a siamese time delay neural network. In *Advances in Neural Information Processing Systems, Denver, Colorado, USA*, pages 737–744, 1993.

[26] Gregory Koch. Siamese neural networks for one-shot image recognition. In *International Conference on Machine Learning, Deep Learning Workshop*, 2015.

[27] Davide Chicco. Siamese neural networks: An overview. *Artificial Neural Networks*, pages 73–94, 2021.

[28] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA*, pages 1199–1208, 2018.

[29] Binyuan Hui, Pengfei Zhu, Qinghua Hu, and Qilong Wang. Self-attention relation network for few-shot learning. In *IEEE International Conference on Multimedia and Expo Workshops, Shanghai, China*, pages 198–203, 2019.

[30] Xiaoxu Li, Jintao Yan, Jijie Wu, Yuxin Liu, Xiaochen Yang, and Zhanyu Ma. Anti-noise relation network for few-shot learning. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, Auckland, New Zealand*, pages 1719–1724, 2020.

[31] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[32] Kenneth H. Rosen. *Discrete Mathematics and Its Applications*. McGraw-Hill Education, 2018.

[33] Bernard De Baets and Etienne Kerre. Fuzzy relations and applications. In *Advances in Electronics and Electron Physics*, volume 89, pages 255–324. Elsevier, 1994.

[34] Ling Zhang and Bo Zhang. *Quotient Space Based Problem Solving: A Theoretical Foundation of Granular Computing*. Tsinghua University Press, 2014.

[35] W. Bandler and L. J. Kohout. Special properties, closures and interiors of crisp and fuzzy relations. *Fuzzy Sets and Systems*, 26(3):317–331, 1988.

[36] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[37] J. Ross Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[38] Yann LeCun, Yoshua Bengio, and Geoffrey E. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[39] Harry Zhang. The optimality of naive bayes. In *International Florida Artificial Intelligence Research Society Conference, Miami Beach, Florida, USA*, pages 562–567, 2004.

[40] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[41] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[42] Ryan M. Rifkin and Aldebaro Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141, 2004.

[43] Thomas M. Cover and Peter E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

[44] Siyuan Shan, Yang Li, and Junier B. Oliva. Meta-neighborhoods. In *Advances in Neural Information Processing Systems, virtual*, pages 5047–5057, 2020.

[45] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations, San Diego, CA, USA*, 2015.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [No]
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See **Appendix** A.3.
   (b) Did you include complete proofs of all theoretical results? [Yes] See **Appendix** A.3.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The code of the proposed method is provided in supplemental material. And the data sets can be download from the links in **Appendix** A.4.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See **Appendix** A.4.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] The code of the proposed method is provided in the supplemental material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No]

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]