International Journal of Approximate Reasoning 143 (2022) 78–101

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

# GLRM: Logical pattern mining in the case of inconsistent data distribution based on multigranulation strategy

Qian Guo [a,c,d], Yuhua Qian [a,b,c,d,*], Xinyan Liang [a,c,d]

[a] Institute of Big Data Science and Industry, Shanxi University, Taiyuan 030006, Shanxi, China
[b] Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, Shanxi University, Taiyuan 030006, Shanxi, China
[c] Engineering Research Center for Machine Vision and Data Mining of Shanxi Province, Taiyuan 030006, Shanxi, China
[d] School of Computer and Information Technology, Shanxi University, Taiyuan 030006, Shanxi, China

ARTICLE INFO

*Article history:*
Received 4 October 2021
Received in revised form 25 December 2021
Accepted 12 January 2022
Available online 19 January 2022

*Keywords:*
Multigranulation
Granular computing
Data-driven logic learning
Open-set Fashion-Logic data set

ABSTRACT

Recently, many learning-based methods have explored logic learning task in the assumption that the training set and testing set are from the consistent distribution, achieving good performance. But, in most cases, this assumption does not hold. In this paper, we explore this topic on the open-set logic reasoning task where the digit length and the sequence length of the training set and testing set are from inconsistent distributions. To address this issue, inspired by multigranulation studies in granular computing, we propose a granulation logic reasoning machine, namely GLRM. In this method, this open-set task is granulated into a series of sub-tasks from two dimensions: the digit length and the sequence length, and then these sub-tasks are conquered one by one. Finally, the results of the sub-tasks are organized into the final result. The effectiveness of GLRM is demonstrated by experiments on the open-set Fashion-Logic data set and the open-set Fashion-Logic task proposed in this paper. This study provides a novel view for solving open-set logic reasoning tasks and promotes the research of data-driven logic learning.

## 1. Introduction

Logic reasoning is a worthwhile task and has been applied into many applications such as medical diagnosis [1,2], group decision making [3], feature selection [4–6], object recognition [7], Multi-Modal Classification [8] and others [9–12]. Some studies mainly focus on designing reasoning patterns/rules for different situations such as rough reasoning [13–17], fuzzy reasoning [18–20], evidential reasoning [21–23], probabilistic reasoning [24–26], quantitative reasoning [27] and bayesian reasoning [28,29]. These reasoning patterns are generally defined by human in advance. In order to automatically mine logic reasoning patterns without relying on experts, [30–32] propose logic learning that aims to directly learn logic pattern from data. Fig. 1 illustrates the difference between human defining logic and logic learning. Logic learning is formalized as follows:

[*] Corresponding author at: Institute of Big Data Science and Industry, Shanxi University, Taiyuan 030006, Shanxi, China.
*E-mail addresses:* czguoqian@163.com (Q. Guo), jinchengqyh@126.com (Y. Qian), liangxinyan48@163.com (X. Liang).

https://doi.org/10.1016/j.ijar.2022.01.005

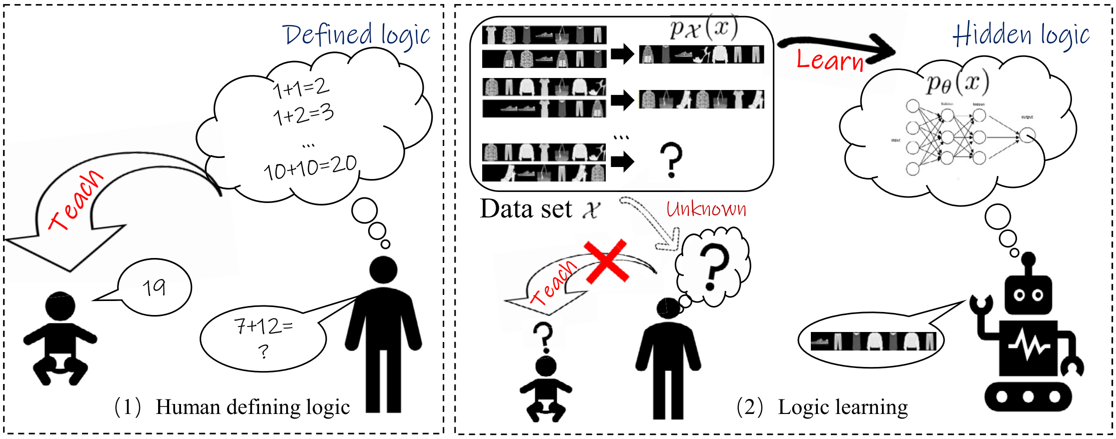


**Fig. 1.** The difference between human defining logic and logic learning. (1) A baby reasons the true answer 19 of an unseen arithmetic operation $7 + 12$ using the addition logic that has been defined by experts. (2) A baby cannot reason the true answer of an unseen operation because the hidden logic has not yet been defined by experts. In logic learning, the hidden logic is encoded in a LRM with a data-driven learning way.

**Definition 1.** (Logic learning): Given a hidden logic data set $\mathcal{X} = \{x_1, x_2, \cdots, x_{|\mathcal{X}|}\}$ that consists of $|\mathcal{X}|$ examples. The aim of the logic learning is to learn a parameterized logic reasoning machine $p_\theta(x)$ with parameter $\theta$ by minimizing the gap between $p_\theta(x)$ and the true data distribution $p_{\mathcal{X}}(x)$. It can be formalized as the following optimization problem.

$$\min_\theta d(p_{\mathcal{X}}(x), p_\theta(x)) \tag{1}$$

where $p_{\mathcal{X}}(x)$ is determined by the data set $\mathcal{X}$. $d(\cdot)$ denotes a measure function of the differences between two probability distributions.

Some tasks do benefit from the reasoning patterns/rules that are provided/defined by domain experts [33,34]. For example, Ivan et al. [35] proposed logic tensor networks by combining neural networks with first-order fuzzy logic to improve the performance of semantic image interpretation. Also, Tran et al. [36] proposed a deep logic network that integrates symbolic representation and quantitative reasoning, demonstrating that the use of logical rules can be beneficial to network performance.

But defining reasoning patterns is time-consuming and laborious. In addition, not all tasks can be predefined reasoning rules, and in some complex systems such as some undeciphered Mayan system, we may not know any reasoning rules in advance and can not define them.

To solve these problems, the data-driven logic learning is proposed to make the machine mine logic pattern automatically. Following the traditional machine learning paradigm (the distribution of the training set is consistent with that of the testing set), a few works have achieved good performance on logic reasoning task. Hoshen et al. [37] used the deep neural network to solve the visual learning task of arithmetic operation in an end-to-end manner, instead of the manner that processing separately perception and cognition subtasks. In order to tackle the difficult logic task, Guo et al. [31] proposed a divide and conquer model, which decomposes the difficult logic task into $k$ sub-tasks according to the principle that the combination difficulty of subtasks is lower than that of original task. But, when this principle can not be met, they may fail in mining logical patterns. Note that the symbols embed in data sets used in [31,37] may be simpler, Guo et al. [30] replaced the number symbols with the symbols from Fashion-MNIST and proposed new benchmark logic data sets, namely Fashion-Logic data set. Based on the proposed data set, they showed that the logic reasoning machines (LRMs) used in [31] still work well on some logic systems that consist of complex symbols.

Despite their success, existing LRMs are trained based on the assumption that the distribution of training set and testing set is consistent. In fact, the open-set tasks are more common for human. In order to further test the performance of the LRMs, the simply open-set tasks where the digit length of the training set and testing set is from inconsistent distributions are proposed. Unfortunately, the performance of the LRMs decreases sharply on the proposed tasks. However, for logic reasoning task, it is more meaningful to improve reasoning ability on open-set task that is more conducive to test the generalization performances and real reasoning abilities of these models. Human beings are very good at dealing with the open-set tasks. Can machines take advantage of human problem-solving ideas to deal with the open-set tasks?

It is generally acknowledged that the ability of decomposing whole task into parts, and then combining the results of parts to obtain whole result plays a fundamental role in the process of human reasoning [38–41]. In existing artificial intelligence technologies, granular computing introduced by Zadeh and Lin [41,42], is able to well simulate such ability of human by three basic processes: information granulation involving decomposition of whole into parts, organization involving integration of parts into whole and causation involving association of causes with effects [41,43,44]. In 1998, Yager and

Filev [45] pointed out that human uses information granularities that are obtained by parting whole to realize their observation, measurement, conceptualization and reasoning. To date, granular computing has achieved great success in artificial intelligence, information processing, data mining and knowledge discovery [46–48]. For example, Zhang et al. investigated multi-source data fusion using multi-granularity and achieved good performance [49,50]. These show that the strategy of decomposing whole into parts does be effective to address complex tasks.

In this paper, a deeper study on logic learning – learning logic from inconsistent distribution data – will be done using the decomposing strategy. Motivated by multigranulation work [48,51,52] in granular computing, a granulation logic reasoning machine, namely GLRM, is developed for mining the logic patterns. Specifically, we take the digit length and the sequence length of the example of the training set as the minimum decomposition unit. According to the minimum decomposition unit, we decompose the complex logical task (open-set task) into a series of sub-tasks. Then we solve each sub-task separately, and finally organize the solutions of these sub-tasks into the final answer.

The main contributions of this paper are summarized as follows:

1. The granulation logic reasoning machine (GLRM) is proposed to address the open-set task. Specially, the open-set task is granulated into a series of sub-tasks from both the digit length and the sequence length dimensions. Then the LRM with good performance is used to solve these sub-tasks. The final result is achieved by organizing the results of these sub-tasks. Extensive experiments demonstrate the superiority of GLRM in reasoning.
2. The open-set Fashion-Logic data set and open-set Fashion-Logic task are proposed for testing the effectiveness of GLRM. Different from traditional machine learning paradigm, the digit length and the sequence length of training set are short, but the digit length and the sequence length of testing set are very long. The open-set Fashion-Logic data set is a very challenging open-set data set.
3. We deeply evaluate the reasoning abilities of the LRMs on the large-scale Fashion-Logic testing set and the simplified version of open-set Fashion-Logic data set. The LRMs have good performances on the testing set that covers more examples in integer space. However, the performance of LRMs becomes worse on the simple open-set tasks. This indicates that LRMs have certain reasoning ability but poor ability on open-set task.

The rest of this paper is organized as follows. In Section 2, the logic learning methods in the case of consistent and inconsistent data distribution, the Fashion-Logic data set and Fashion-Logic task are reviewed. In Section 3, we deeply assess the reasoning performances of the LRMs on the large-scale Fashion-Logic testing set and the simplified version of open-set Fashion-Logic data set. In Section 4, we detail the open-set Fashion-Logic data set and the open-set Fashion-Logic task and propose the GLRM to deal with this task. In Section 5, the detailed results of effectiveness studies and analyses of GLRM are reported. Finally, we conclude and indicate several issues for future work.

## 2. Related work

In this section, we review the logic learning methods in the case of consistent and inconsistent data distribution, the Fashion-Logic data set and the Fashion-Logic task.

### *2.1. Logic learning methods in the case of consistent data distribution*

**Visual learning of arithmetic operation**: In the general case, visual learning of arithmetic operation task is divided into two sub-tasks: perception and cognition, and then the two sub-tasks are learned separately. Unlike previous work, Hoshen et al. [37] explored whether a machine can end-to-end learn arithmetic operations from images. Following the traditional machine learning paradigm, they generated three kinds of arithmetic operation (addition, subtraction and multiplication) data sets. That is to say, the distribution of training set of these data sets is consistent with that of the corresponding testing set. For each example, there are two input images and one output image, each image is embedded with a 7-digit number. Given the testing input images, the goal of [37] was to predict the output image of the decimal operation of the input images. [37] used an end-to-end learning method to replace the learning method in which subtasks were processed separately, achieving good results on the visual learning of arithmetic operation.

**Mining logic patterns**: In some complex systems such as undeciphered Mayan systems, human beings do not know any reasoning rules in advance. Therefore, Guo et al. [30] proposed a learning-based method for directly mining logical patterns from the visual data that consist of complex symbols. Following the traditional machine learning paradigm, they generated the logic relation data set named Fashion-Logic (described in detail in Section 2.3). That is to say, the distribution of training set is consistent with that of the corresponding testing set. Specifically, there are two kinds of logic relations hidden in the Fashion-Logic data set: bitwise logic (Bitwise And and Bitwise Or) and arithmetic logic (Addition and Subtraction). For each example, there are two input images and one output image, each image is embedded by a series of figure symbols from the Fashion-MNIST data set [53]. In their experiments, the tested machine does not know the meanings of the figure symbols embedded in the images and the logic relation between input and output images. Given the testing input images, the goal of [30] was to predict the output image of the bitwise logic (or arithmetic logic) of input images. [30] conducted a deep exploration and analysis, showing the feasibility of directly mining logical patterns from the visual data that consist of complex symbols.

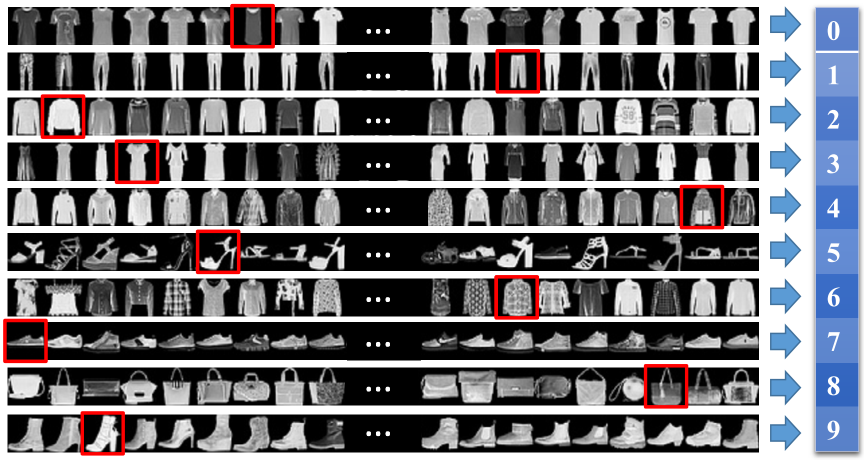


**Fig. 2.** Some examples of Fashion-MNIST data set. (For interpretation of the colours in the figure(s), the reader is referred to the web version of this article.)

Under the premise that the distribution of the training set is consistent with that of the testing set, these methods have explored and proved that logic patterns can be directly learned from the visual data.

### 2.2. Logic learning methods in the case of inconsistent data distribution

**Abductive learning** [54,55]: Unlike the human problem-solving process, in most current machine learning systems, perception and reasoning modules can not be learned simultaneously. Noting this phenomenon, Zhou et al. proposed the abductive learning that unifies the perception and reasoning into a learning framework. The Digital Binary Additive (DBA) and the Random Binary Additive (RBA) were generated to verify the validity of the abductive learning framework. Each example consists of an image that is embedded with an equation. The equations embedded in the training set are the short binary addition logic, and the equations embedded in the testing set are the long binary addition logic. This means that the distribution of training set is inconsistent with that of the corresponding testing set on DBA and RBA, which is different from the traditional machine learning paradigm. The goal of the abductive learning framework was to judge whether the equation embedded in the given image is true or not. It is worth noting that the abductive learning framework can predict whether the long equations are correct after learning on the short equations.

Under the premise that the distribution of the training set is inconsistent with that of the testing set, Zhou et al. have explored that the machine can directly judge whether the binary addition equations embedded in images are true or not.

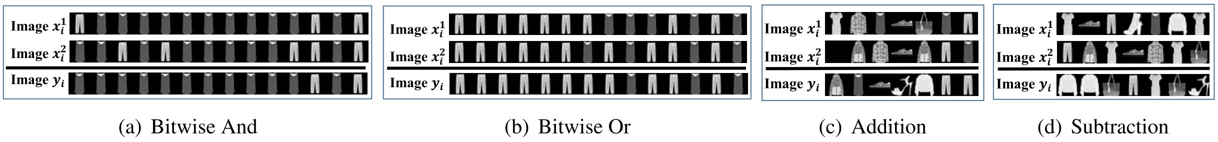
In summary, Sections 2.1 and 2.2 have carried out some meaningful explorations and analyses for logic learning, especially in the case of consistent data distribution, but the study of inconsistent data distribution is relatively few and simple. However, the phenomenon of inconsistent data distribution is widespread in the real world, and the logic learning in the case of inconsistent data distribution is one of the most urgent problems in the field of artificial intelligence. Therefore, our work makes a relatively deep study of logic learning in the case of inconsistent data distribution and provides an effective solution.

### 2.3. Fashion-Logic data set and Fashion-Logic task

In [30], the Fashion-Logic data set is embedded by special symbols and contains some logic relations. Specifically, four logic relations: Bitwise And, Bitwise Or, Addition and Subtraction are adopted and these logic relations are often used by the related researches to logic such as [56] and [57]. The special symbols embedded in the Fashion-Logic data set are from the examples of Fashion-MNIST data set [53]. Here we briefly review the construction process of the Fashion-Logic data set. First, the logic operations were generated, and then were replaced with the corresponding figure symbols[1] that were randomly selected from each class of Fashion-MNIST (it is noted that digits 0, 1, ... 9 only were replaced with the images marked with red boxes in Fig. 2). Some examples of the Fashion-MNIST data set are illustrated in Fig. 3. For Bitwise And and Bitwise Or logic relations, a maximum of 14 digits are embedded in an image. For Addition and Subtraction logic relations, a maximum of 7 digits are embedded in an image. Each logic relation consists of 50,000 training examples, 5,000 validation examples and 5,000 testing examples. This ensures that the proportion of numbers used for training is a very small fraction of all possible combinations. The testing examples are not included in the training or validation examples.

Based to the Fashion-Logic data set, Guo et al. [30] proposed a visual logic reasoning task called the Fashion-Logic task where a learning machine directly mines and reasons the relation between two input images and one output image. The

[1] The term “figure symbol” for what is commonly referred to as “image” in “Fashion-image” from to differentiate from the term “image”.



(a) Bitwise And (b) Bitwise Or (c) Addition (d) Subtraction

**Fig. 3.** The examples of the Fashion-Logic data set. The logic relation hidden in 3(a), 3(b), 3(c) and 3(d) are 1000000000101&00101000001101=00000000000101, 11111110001010|11111101011010=11111111011010, 3607801+467410=4075211 and 3719023-1437638=2281385, respectively.

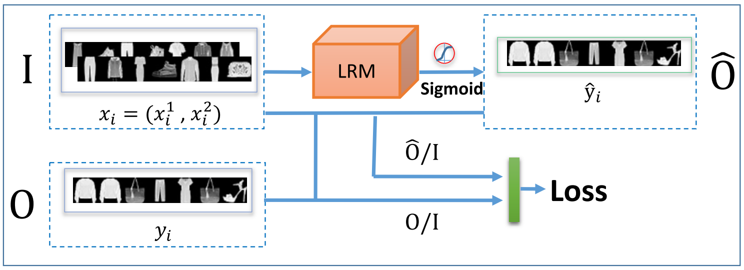


**Fig. 4.** The general workflow of the Fashion-Logic task.

learning machine does not know the meanings of the figure symbols embedded in images and logic relation among images in advance. This task is defined as follows.

**Definition 2.** (Fashion-Logic task): In the Fashion-Logic task, given a data set $D = \{(x_i, y_i)\}_{i=1}^N$, where $x^{(1)}$ and $x^{(2)}$ in $x_i$ are the input images and $y_i$ is the output image where each pixel values in each image are normalized into $0 \sim 1$. This task mines a mapping LRM from the input space $I = \{x_i\}_{i=1}^N$ to the output space $O = \{y_i\}_{i=1}^N$, i.e., $\text{LRM} : I \mapsto O$. When two unseen input images $x^{(1)}$ and $x^{(2)}$ are given, LRM is able to output the logical pattern $y$ between both of them, i.e., $y = LRM(x^{(1)}, x^{(2)})$.

The general workflow of the Fashion-Logic task is shown in Fig. 4, where $I$ is the input, $O$ is the ground-truth, $\hat{O}$ is the logic relation results reasoned by $f(\text{LRM}_W(x_i^1, x_i^2))$. LRM is a logic reasoning machine that can be implemented using different learning models. Hence, the Fashion-Logic task can be used to test the logic reasoning ability of the different learning models. In [30], the logic reasoning ability of some LRMs such as CNN-LSTM, MLP, CNN-MLP, Autoencoder and ResNets, has been tested on this task when the distribution of the training set is consistent with that of the testing set.

More specifically, [30] transforms this task into a regression problem with the mean square error (MSE) loss function. It can be written as the following optimization problem.

$$\begin{aligned} W^* &= \arg\min_W MSE(f(\text{LRM}_W(I)), O) \\ &= \arg\min_W \frac{1}{N}\sum_{i=1}^N \text{MSE}(f(\text{LRM}_W(x_i^{(1)}, x_i^{(2)})), y_i) \\ &= \arg\min_W \frac{1}{N}\sum_{i=1}^N \sqrt{\sum_{k=1}^K (f(\text{LRM}_W(x_i^{(1)}, x_i^{(2)}))_k - y_{ik})^2}, \end{aligned} \tag{2}$$

where the activation function $f$ is a sigmoid function and LRM is a logic reasoning machine that is parameterized by $W$. Eq. (2) is differentiable with respect to the parameter $W$, and can be efficiently solved by using the gradient descent method.

## 3. Do the LRMs work in larger integer space and the open-set logic reasoning tasks?

In the existing works, the testing set often covers a few examples in integer space. Therefore, we further assess the performance of the LRMs used in [30] on the large-scale Fashion-Logic testing set that covers more examples in integer space. The difference between Fashion-Logic testing set and large-scale Fashion-Logic testing set is marked with red box in Fig. 5.

On the other hand, the digit length and the sequence length of testing set are often consistent with those of training set in the existing works. Therefore, we further assessed the performance of the LRMs on two simple open-set tasks where the digit length of testing set is different from that of training set. **Task 1:** Training on the digit length of mainly 6 and 7 then testing on the digit length of 1∼5. **Task 2:** Training on the digit length of 1∼5 then testing on the digit length of 6∼10.

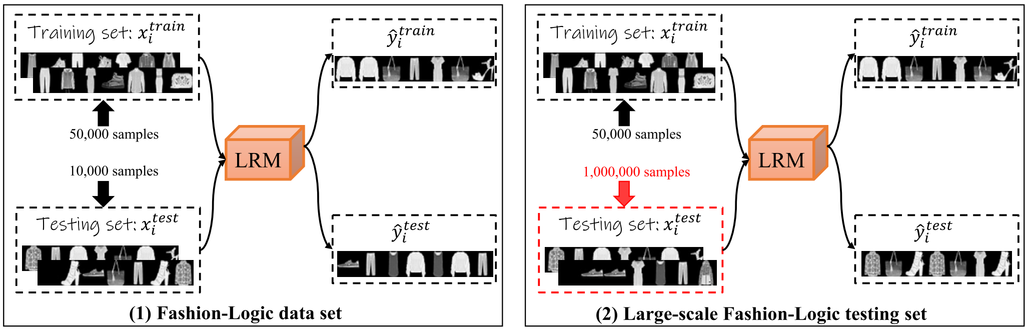

Training set: $x_i^{train}$
50,000 samples
10,000 samples
Testing set: $x_i^{test}$
LRM
$\hat{y}_i^{train}$
$\hat{y}_i^{test}$
(1) Fashion-Logic data set
Training set: $x_i^{train}$
50,000 samples
1,000,000 samples
Testing set: $x_i^{test}$
LRM
$\hat{y}_i^{train}$
$\hat{y}_i^{test}$
(2) Large-scale Fashion-Logic testing set

**Fig. 5.** The difference between Fashion-Logic testing set and large-scale Fashion-Logic testing set.

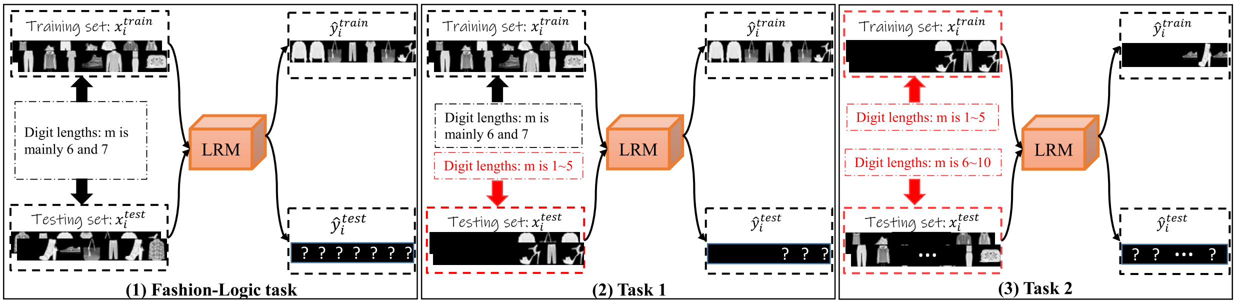


**Fig. 6.** The differences among the Fashion-Logic task, task 1 and task 2.

**Table 1**
The performance on the large-scale Fashion-Logic testing set.

| LRM | Logic relations | | | |
|---|---|---|---|---|
| | Bitwise And | Bitwise Or | Addition | Subtraction |
| Random conjecture | $\frac{1}{2^{14}}$ | $\frac{1}{2^{14}}$ | $\frac{1}{10^7}$ | $\frac{1}{10^7}$ |
| CNN-LSTM | 100% | 100% | 54.79% | 52.23% |
| MLP | 100% | 100% | 98.43% | 98.35% |
| CNN-MLP | 100% | 100% | 99.81% | 99.73% |
| Autoencoder | 100% | 100% | 97.46% | 97.35% |
| ResNet18 | 100% | 100% | 99.98% | 99.97% |
| ResNet50 | 100% | 100% | 99.98% | 100% |
| ResNet152 | 100% | 100% | 100% | 100% |

Taking addition logic as an example, the differences among the Fashion-Logic task, task 1 and task 2 are marked with red boxes in Fig. 6.

The experimental setups and the structures of the tested LRMs are given in the Appendix.

### 3.1. Performance analysis on the large-scale Fashion-Logic testing set

This subsection aims to analyze the performance of the LRMs described in the Appendix on the larger-scale Fashion-Logic testing set. Specifically, following the same way as generating the Fashion-Logic data set, we generate one million testing examples (large-scale Fashion-Logic testing set) without repetition whose scale is 20 times larger than that of the training set (50,000) of the Fashion-Logic data set on each logic relation. For each predicted image, it is correct if and only if all the embedded figure symbols are right. The experimental results are shown in Table 1.

From Table 1, one can get four observations.

(1) For all logic relations, the performances of all the LRMs are far better than the performances of random conjectures, which shows that data-driven logic mining is feasible.
(2) The accuracies of all the LRMs except for CNN-LSTM are higher than 97.00%. This indicates that these LRMs trained by very small training set (the training set only occupy a very small percentage in all possibility), can well mine logic patterns from the unseen data.
(3) The performances of all the LRMs on the Bitwise And and Bitwise Or are better than those on the Addition and Subtraction. This is consistent with human perception: the bitwise logic is easier than the arithmetic logic.

**Table 2**
The testing accuracies on task 1.

| Logic relation | $m$ | CNN-LSTM | MLP | CNN-MLP | Autoencoder | ResNet18 | ResNet50 | ResNet152 |
|---|---|---|---|---|---|---|---|---|
| Addition | 1 | 0% | 0% | 0% | 100% | 100% | 100% | 100% |
| | 2 | 6.86% | 0% | 0% | 100% | 100% | 100% | 100% |
| | 3 | 27.44% | 0.11% | 7.78% | 99.89% | 100% | 100% | 100% |
| | 4 | 52.96% | 72.79% | 83.67% | 98.93% | 100% | 100% | 100% |
| | 5 | 54.04% | 95.46% | 97.55% | 98.64% | 100% | 100% | 100% |
| Subtraction | 1 | 60.00% | 0% | 100% | 100% | 100% | 100% | 100% |
| | 2 | 15.56% | 2.22% | 71.11% | 100% | 100% | 100% | 100% |
| | 3 | 60.00% | 58.44% | 87.56% | 99.56% | 100% | 100% | 100% |
| | 4 | 59.60% | 82.90% | 96.00% | 99.70% | 100% | 100% | 100% |
| | 5 | 57.03% | 97.14% | 98.09% | 98.83% | 99.97% | 100% | 100% |

(4) For the Addition and Subtraction logic relations, the performance of CNN-LSTM is lower than that of others. This is due to the structure of the CNN-LSTM: the CNN-LSTM processes the input images sequentially. As a result, the global structure information can not be considered in the learning reasoning process. However, the carry operation of addition (the borrow operation of subtraction) requires interaction between the input images to get the correct result. Therefore, other LRMs are more suitable for the Addition and Subtraction. The relative analysis and observation suggest that the global structure information plays a significant role in the logic reasoning learning task.

The proportion of numbers used for training is a very small fraction of all possible combinations, which means that the LRMs have seen very few examples. Based on the results in Table 1, the LRMs have obtained certain logic reasoning abilities when they work according to the workflow of the traditional machine learning, i.e., a LRM is good at generalizing the unseen large-scale Fashion-Logic testing set that has the consistent distribution with training set.

### *3.2. Performance analysis on two simple open-set tasks*

The subsection aims to test the performance of the LRMs on Task 1 and Task 2. For our research purpose, two kinds of the simplified version of open-set Fashion-Logic data set for task 1 and task 2 are generated, respectively. For convenience, we use $m$ to represent the digit length of each number embedded in each image. Next, tasks 1 and task 2 are described in detail.

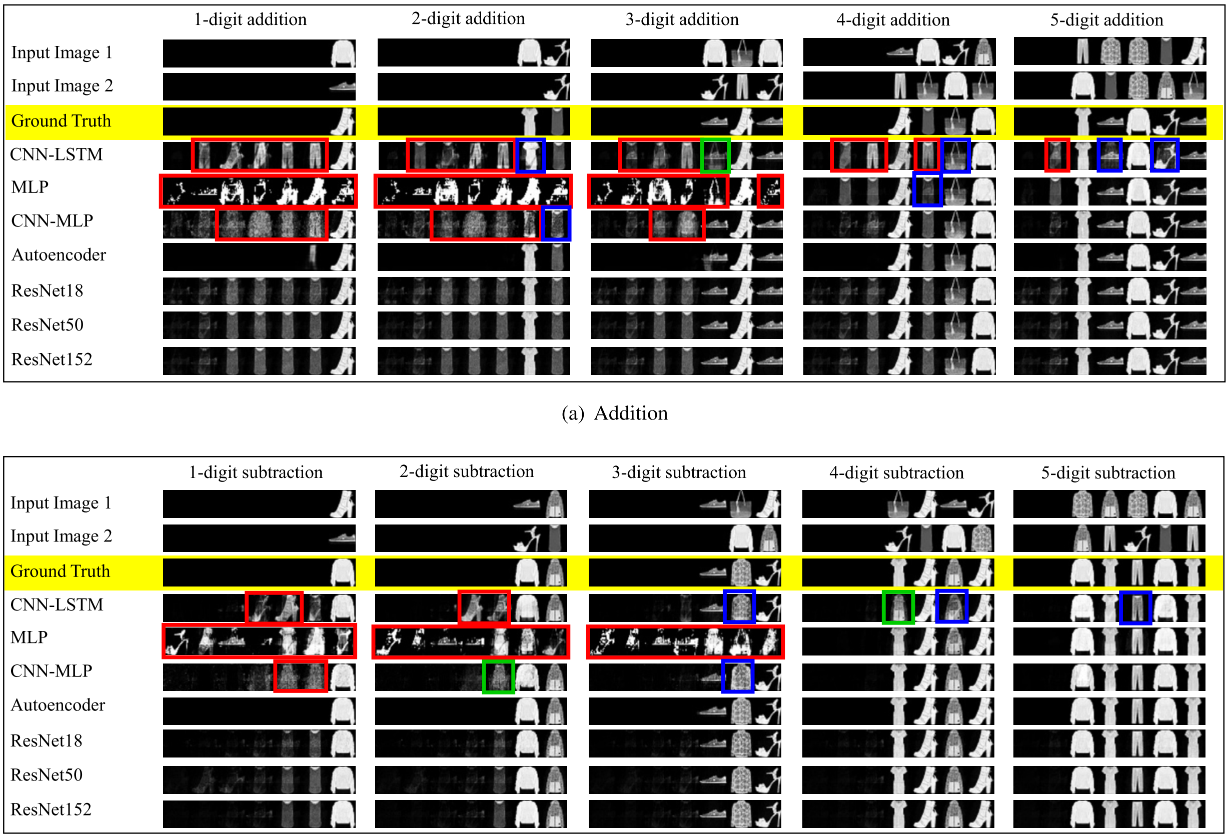
**Task 1**: For task 1, a simplified version of open-set Fashion-Logic data set is generated, which includes the training set with the digit lengths mainly 6 and 7 and the testing set with the digit lengths of 1∼5. In fact, the data distribution of the training set of this simplified version of open-set Fashion-Logic and the Fashion-Logic is consistent. We directly take the training set of the Fashion-Logic as the training set of this simplified version of open-set Fashion-Logic. The performances are shown in Table 2 and Fig. 7.

Table 2 shows that Autoencoder and ResNets have very good performances. Moreover, the greater $m$ is, the better the performances of most LRMs are. For example, the accuracies of MLP and CNN-MLP are 0% when $m = 1, 2$, they achieve the accuracies of 95.46% and 97.55% when $m = 5$ on Addition. These results suggest that: (1) The performances of some LRMs may be affected by the testing set whose distribution is considerably different from that of training set. (2) Some LRMs indeed have a better generalization in this setting. Some visual effects are shown in Fig. 7.

The data distribution analysis is shown in Fig. 8. The first two rows in Fig. 8 show that the examples from the training set and the testing set have the consistent data distribution in term of the digit length (mainly 6 and 7) because of adopting the same sampling strategy. Under this data distribution, all LRMs get good performances (already analyzed in Subsection 3.1). Because we directly take the training set of the Fashion-Logic as the training set of this simplified version of open-set Fashion-Logic, in task 1, the first row and third row in Fig. 8 show that the examples from this simplified version of open-set Fashion-Logic training set and testing set. Obviously, in terms of digit length, the data distribution of this simplified version of open-set Fashion-Logic data set is inconsistent. For each digit length, as shown in the last row in Fig. 8, the testing examples used in this task have about the same occupancy ratios with logarithmic scale, i.e., $\frac{\log(\text{Used})}{\log(\text{Total})}$.

**Task 2**: For task 2, the LRMs are trained with the examples whose digit lengths are 1∼5, while their performances are evaluated with the examples whose digit lengths are 6∼10. We generate another simplified version of open-set Fashion-Logic data set, which includes the training set with the digit lengths of 1∼5 and the testing set with the digit lengths of 6∼10. Table 3 shows that all LRMs but Autoencoder fail in this case. This indicates that the performances of these LRMs have a limitation in generalization task from short digit to long digit that human is very good at.

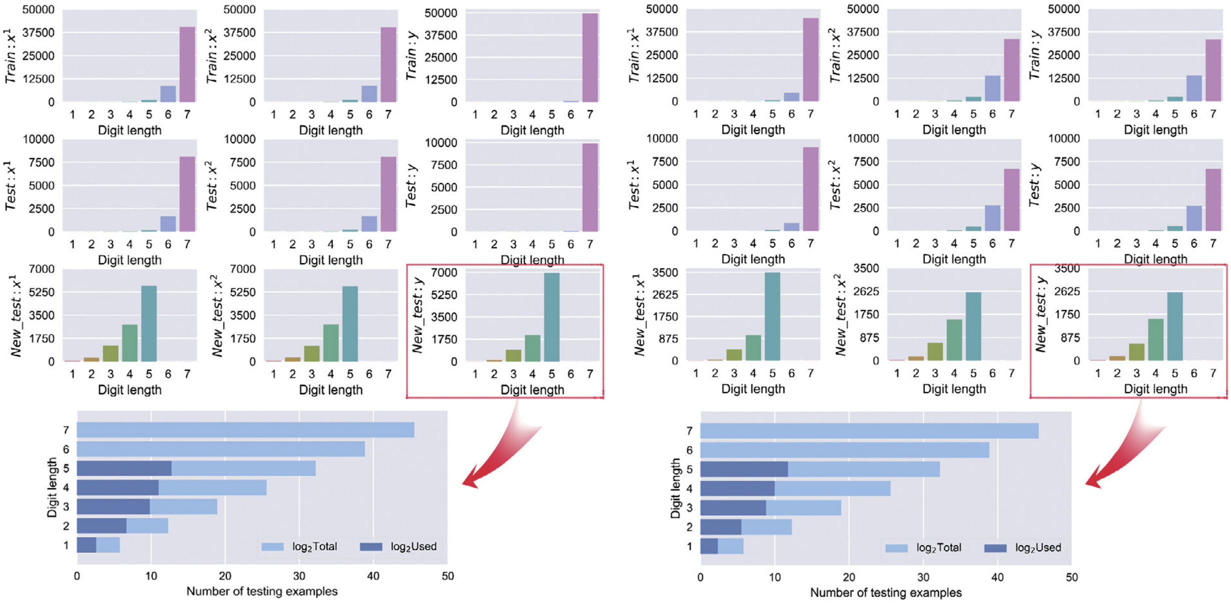
Based on the above analyses, it is concluded that on the one hand, these LRMs have very good performances when the distribution of the training set is consistent with that of the testing set. The learning manner is consistent with the machine learning paradigm. Whereas their reasoning abilities become worse even lose when the distribution of the training set is inconsistent with that of the testing set. The learning manner of task 1 and task 2 is beyond the machine learning paradigm. However, this ability is generally acknowledged to be an essential feature of human intelligence in reasoning. In what follows, we will address this problem.



(a) Addition

(b) Subtraction

**Fig. 7.** The visual effects on task 1. The blurry but right figure symbols predicted are marked with blue boxes, blurry and wrong those are marked with green boxes, and wrong those are marked with red boxes.



(a) Addition

(b) Subtraction

**Fig. 8.** The data distribution analysis on task 1.

**Table 3**
The testing accuracies on task 2.

| Logic relation | $m$ | CNN-LSTM | MLP | CNN-MLP | Autoencoder | ResNet18 | ResNet50 | ResNet152 |
|---|---|---|---|---|---|---|---|---|
| Addition | 6 | 0% | 0% | 0% | 98.3% | 0% | 0% | 0% |
| | 7 | 0% | 0% | 0% | 97.2% | 0% | 0% | 0% |
| | 8 | 0% | 0% | 0% | 96.8% | 0% | 0% | 0% |
| | 9 | 0% | 0% | 0% | 95.0% | 0% | 0% | 0% |
| | 10 | 0% | 0% | 0% | 94.5% | 0% | 0% | 0% |
| Subtraction | 6 | 0% | 12.3% | 11.8% | 98.2% | 22.9% | 26.5% | 26.2% |
| | 7 | 0% | 1.7% | 1.5% | 97.2% | 2.6% | 3.0% | 2.9% |
| | 8 | 0% | 0% | 0.1% | 97.2% | 0.2% | 0.3% | 0.3% |
| | 9 | 0% | 0.1% | 0.1% | 96.5% | 0.1% | 0.1% | 0.1% |
| | 10 | 0% | 0% | 0% | 95.1% | 0% | 0% | 0% |

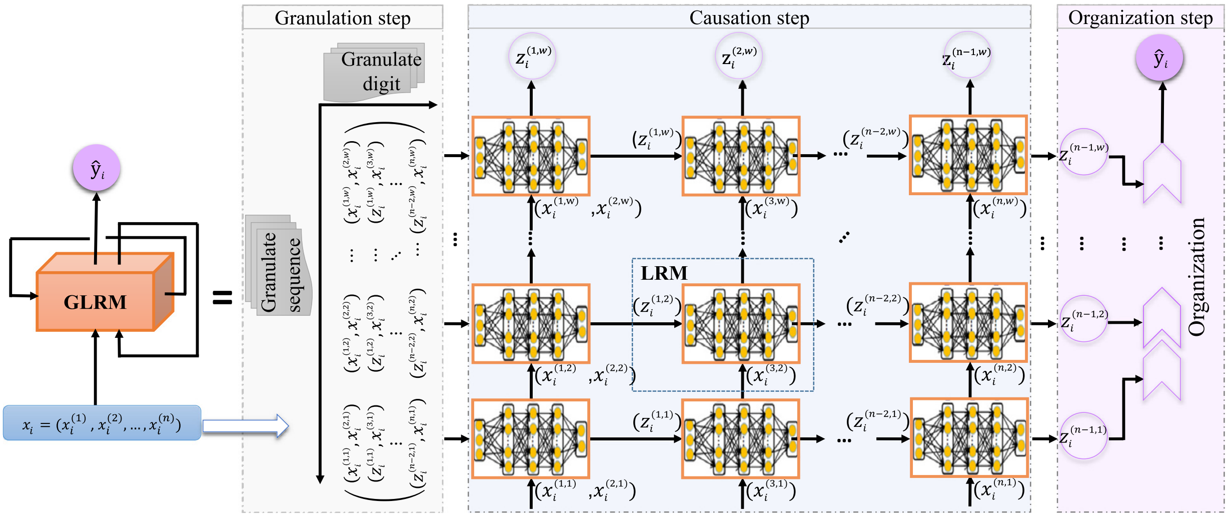


**Fig. 9.** Overview of the proposed GLRM.

## 4. Granulation logic reasoning machine

The challenge of open-set logic reasoning task lies in a fact that digit length and sequence length of the testing examples are often different from those of training examples. For example, after being taught that $1+1=2$ and $2+5=7$, how do humans calculate the result of the formula $1+1+5=$? First, humans would calculate $1+1=2$, and then $2+5=7$. In other words, humans usually decompose the open-set logic reasoning task into a few known sub-tasks, then obtain the final result by solving them one by one with the learned logic rules.

To make machines solve open-set logic reasoning task like humans, we propose a granulation logic reasoning machine (GLRM) that is inspired by granular computing that simulates the learning ability of humans [41,58–60]. GLRM is able to mine the logic patterns of the open-set testing set without being further trained. The overview of the proposed GLRM is illustrated in Fig. 9. Similar to granular computing, GLRM also includes three steps: granulation, causation and organization. To conveniently describe below, we give some notations shown in Table 4.

**Granulation step**: Given an image sequence example $x_i = (x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(n)})$, where $n$ denotes the sequence length of $x_i$, $x_i^{(j)}$ denotes the $j$-th image in the sequence $x_i$ and its digit length is denoted as $m$, i.e., it contains $m$ figure symbols. Obviously, a LRM will fail when $n$ or $m$ of testing set is greater than that of training set. In order to address this issue, motivated by multigranulation work [51,52,61–65] in granular computing, we granulate $x_i$ into easier subimages from two dimensions: the sequence length and the digit length.

Specifically, $x_i$ is first granulated from the dimension of the sequence length by the following way:

$$G_s(x_i) = \{(x_i^{(1)}, x_i^{(2)}), (z_i^{(1)}, x_i^{(3)}), \dots, (z_i^{(n-2)}, x_i^{(n)})\}, \tag{3}$$

where $z_i^{(j)}$ denotes the logic relation among all images in the subsequence $x_i^{(1:j+1)}$ that consists of the previous $j+1$ images in $x_i$. It can be further denoted as the logic relation between $(j+1)$-th image $x_i^{(j+1)}$ and $z_i^{(j-1)}$. Following this way, we can obtain it using a well-trained LRM according to the following recursion way:

**Table 4**
Notations.

| Symbol | Explanation |
|---|---|
| $x_i$ | The $i$-th image sequence example in the open-set data set |
| $n$ | The sequence length, i.e., the number of the images that each sequence contains |
| $m$ | The digit length, i.e., the number of the figure symbols that each image contains |
| $k$ | The window size of function Win |
| $s$ | The stride of function Win |
| $x_i^{(j)}$ | The $j$-th image in the sequence $x_i$ |
| $z_i^{(j)}$ | The logic relation among images from $x_i^{(1)}$ to $x_i^{(j+1)}$ |
| $x_i^{(j,l)}$ | A subsequence of $x_i^j$ consisting of the figure symbols in $x_i^{(j)}$ whose indexes are from $(l-1)s+1$ to $(l-1)s+k$ |
| $z_i^{(j,l)}$ | A subsequence of $z_i^j$ consisting of the figure symbols in $z_i^{(j)}$ whose indexes are from $(l-1)s+1$ to $(l-1)s+k$ |

$$z_i^j = \begin{cases} \mathrm{LRM}(x_i^{(1)}, x_i^{(2)}), & j = 1, \\ \mathrm{LRM}(z_i^{(j-1)}, x_i^{(j+1)}), & 1 < j \le n-1, \end{cases} \tag{4}$$

where LRM is a logic reasoning machine which can be one of the well-trained typical deep neural networks such as CNN-LSTM, MLP, CNN-MLP, Autoencoder and ResNets. At the granularity level that is generated from the sequence dimension, the LRM with better reasoning performance can reason the logic relation between $z_i^{(j-1)}$ and $x_i^{(j+1)}$ when the digit length of training examples is equal or greater than that of testing examples, i.e., $m_{train} \ge m_{test}$. But, due to the complexity in the digit dimension, a LRM, even with 100% accuracy, cannot reason the logic relation between $z_i^{(j-1)}$ and $x_i^{(j+1)}$ when $m_{train} < m_{test}$.

Next, for any element $(z_i^{(j-1)}, x_i^{(j+1)}) \in G_s(x_i)$ ($z_i^{(j-1=0)} = x_i^{(1)}$ if $j=1$) in $G_s(x_i)$, denoted as $c_i^{(j)}$, it is further granulated from the dimension of the digit length via a window function as follows.

$$G_d(c_i^{(j)}) = \mathrm{Win}(c_i^{(j)}; k, s) = \mathrm{Win}((z_i^{(j-1)}, x_i^{(j+1)}); k, s), \tag{5}$$

where the function $\mathrm{Win}(c_i^{(j)}; k, s)$ means using a one-dimensional window with the windows size $k$ and stride $s$ to perform a granulating operation over $z_i^{(j-1)} \in c_i^{(j)}$ and $x_i^{(j+1)} \in c_i^{(j)}$. The parameter $s$ controls the amount of communication information when two information granules are organized. The information between both of them can be communicated when $1 \le s < k$. The information between them can not be communicated when $s \ge k$. With this granular function, $c_i^{(j)}$ will be granulated into $\lceil \frac{m-k}{s} \rceil + 1$ parts. In this way, $G_d(c_i^{(j)})$ is denoted as

$$G_d(c_i^{(j)}) = \begin{cases} \{(x_i^{(1,l)}, x_i^{(2,l)})\}_{l=1}^{\lceil \frac{m-k}{s} \rceil + 1}, & j = 1, \\ \{(z_i^{(j-1,l)}, x_i^{(j+1,l)})\}_{l=1}^{\lceil \frac{m-k}{s} \rceil + 1}, & 1 < j \le n-1, \end{cases} \tag{6}$$

where $z_i^{(j-1,l)}$ denotes the logic relation between the output $z_i^{(j-2,l)}$ of $(x_i^{(1,l)}, x_i^{(2,l)}, ..., x_i^{(j-1,l)})$ and a new input image $x_i^{(j,l)}$, and it can be reasoned via:

$$z_i^{(j-1,l)} = \mathrm{LRM}(z_i^{(j-2,l)}, x_i^{(j,l)}), 1 \le l \le \lceil \frac{m-k}{s} \rceil + 1. \tag{7}$$

At the granularity level, the logic relation between $z_i^{(j-1,l)}$ and $x_i^{(j+1,l)}$ is able to be reasoned by the LRM.

After the above steps, we obtain the finer granularity results $G_{s,d}(x_i)$ of $x_i$ than $G_s(x_i)$ as follows:

$$\begin{aligned} G_{s,d}(x_i) &= G_d(G_s(x_i)) \\ &= \{G_d(c_i^{(1)}); G_d(c_i^{(2)}); \ldots; G_d(c_i^{(j)})\} \\ &= \begin{pmatrix} (x_i^{(1,1)}, x_i^{(2,1)}) & (z_i^{(1,1)}, x_i^{(3,1)}) & \cdots & (z_i^{(n-2,1)}, x_i^{(n,1)}) \\ (x_i^{(1,2)}, x_i^{(2,2)}) & (z_i^{(1,2)}, x_i^{(3,2)}) & \cdots & (z_i^{(n-2,2)}, x_i^{(n,2)}) \\ \vdots & \vdots & \ddots & \vdots \\ (x_i^{(1,w)}, x_i^{(2,w)}) & (z_i^{(1,w)}, x_i^{(3,w)}) & \cdots & (z_i^{(n-2,w)}, x_i^{(n,w)}) \end{pmatrix}, \end{aligned} \tag{8}$$

where $w = \lceil \frac{m-k}{s} \rceil + 1$. $G_{s,d}(x_i)$ is called a granular structure [18], and any element $(\cdot,\cdot)$ in it (called a information granule) is a pair of images each of which contains $k$ figure symbols.

**Causation step**: The aim of this step is to reason the hidden logic between image pairs $(\cdot,\cdot)$ in $G_{s,d}(x_i)$. After granulation step, the complex input $x_i$ has been decomposed into easier granulated subimages among which a LRM is able to reason the logic relation as the experimental results shown in Section 3.1.

**Algorithm 1** Pseudo-code of GLRM.

```
Input:  $(X_{train}, Y)$: open-set Fashion-Logic training set;
        $X_{test}$: open-set Fashion-Logic testing set;
        $\mathfrak{L}$: a LRM;
        $k$: the window size of the function Win;
        $s$: the stride of the function Win;
        $t$: the number of the figure symbols that the left segment will contain.
Output: $Y^*$: Reasoned logic for $X_{test}$.
Induce a logic reasoning function LRM based on $(X_{train}, Y)$: LRM $\hookleftarrow \mathfrak{L}(X_{train}, Y)$;
$Y^* \leftarrow \phi$;
for each $x_i \in X_{test}$ do
    Compute $G_s(x_i)$ based on Eq. (3);
    for each $c_i^j \in G_s(x_i)$ do
        Compute $G_d(c_i^{(j)})$ with the trained LRM based on Eq. (5);
    end for
    Form $G_{s,d}(x_i)$ based on $G_d(c_i^{(j)})$ according to Eq. (8);
    Compute $\hat{y}_i^{(l=\lceil \frac{m-k}{s} \rceil+1)}$ based on Eq. (10);
    $\hat{y}_i = \hat{y}_i^{(\lceil \frac{m-k}{s} \rceil+1)}$;
    $Y^* \leftarrow Y^* \cup \{\hat{y}_i\}$;
end for
Return $Y^*$.
```

**Organization step**: The aim of this step is to organize the granulated subimages $\{z_i^{(n-1,l)} = \text{LRM}(z_i^{(n-2,l)}, x_i^{(n,l)})\}_{l=1}^{w}$ to form the final whole $\hat{y}$ that denotes the predicted logic relation among all images in the sequence $x_i$. To this end, we first need to introduce a fusion function for combining two granulated subimages based on a crossover operation variant in evolutionary computation [66]. Specifically, given two granulated subimages denoted as $a = a_{q1} \ldots a_2 a_1$ and $b = b_{q2} \ldots b_2 b_1$ where $q1$ and $q2$ are the number of figure symbols that they contain. First, we need to choose crossover points for $a$ and $b$ which are determined by two parameters $s$ and $t$. Based on the two parameters, their crossover points are $q1 + s - t$ and $q2 - t + 1$, respectively. With these two crossover points, the fusion of $a$ and $b$ is performed by combining the right segment of $a$ (denoted as $a_{q1+s-t} \ldots a_2 a_1$) and the left segment of $b$ (denoted as $b_{q2} b_{q2-1} \ldots b_{q2-t+1}$). Formally, the function is defined as follows.

$$\text{F}(a, b; s, t) = b_{q2} b_{q2-1} \ldots b_{q2-t+1} a_{q1+s-t} \ldots a_2 a_1, \tag{9}$$

where $s$ is a constant that is set as the same value as the parameter stride $s$ of $\text{Win}(c_i^{(j)}; k, s)$. $t$ is the number of the figure symbols that the left segment of $b$ will contain.

With Eq. (9), we can organize all granulated subimages in $\{z_i^{(n-1,l)}\}_{l=1}^{w}$ one by one according to the following recursion way.

$$\hat{y_i}^l = \begin{cases} z_i^{(n-1,1)}, & l = 1, \\ \text{F}(y_i^{(l-1)}, z_i^{(n-1,l)}; s, t), & 1 < l \leq \lceil \frac{m-k}{s} \rceil + 1, \end{cases} \tag{10}$$

where $z_i^{(n-1,l)} = \text{LRM}(z_i^{(n-2,l)}, x_i^{(n,l)})$.

With Eq. (10), when $l = \lceil \frac{m-k}{s} \rceil + 1$, we obtain the final organized element $\hat{y}_i^{(\lceil \frac{m-k}{s} \rceil+1)}$, which is regarded as the reasoned logic (denoted by $\hat{y}_i$) among all images in the sequence $x_i$, i.e., $\hat{y}_i = \hat{y}_i^{(\lceil \frac{m-k}{s} \rceil+1)}$.

The detail process of GLRM is shown in Algorithm 1.

## 5. Effectiveness studies and analyses of GLRM

In this section, the open-set Fashion-Logic data set and open-set Fashion-Logic task are proposed to validate the effectiveness of GLRM. Specifically, compared with the Fashion-Logic testing set, the open-set Fashion-Logic testing set includes three cases of inconsistent data distribution: long sequence (see Section 5.2 for details), long digit (see Section 5.3 for details), and both long sequence and long digit (see Section 5.4 for details). The differences between the Fashion-Logic testing set, the open-set Fashion-Logic testing set with long sequence, the open-set Fashion-Logic testing set with long digit and the open-set Fashion-Logic testing set with both long sequence and long digit are marked with red boxes and shown in Fig. 10. For readability, we still use $m$ and $n$ to represent the digit length and sequence length of each number embedded in each image.

The LRM used in the GLRM is a logic reasoning machine and can be any deep neural network. Because we directly take the training set of the Fashion-Logic as the training set of the open-set Fashion-Logic (see Section 5.1 for details), we use the trained LRMs from Section 3.1 as the LRMs of GLRMs without loss of generality. The trained LRMs have been trained on the open-set Fashion-Logic training set, i.e., $m = 14$ for Bitwise And and Bitwise Or, $m$ is mainly 6 and 7 for Addition and Substraction, and $n = 2$ for all. In the Section 5.2-5.4, the GLRMs are directly tested without further being trained on the

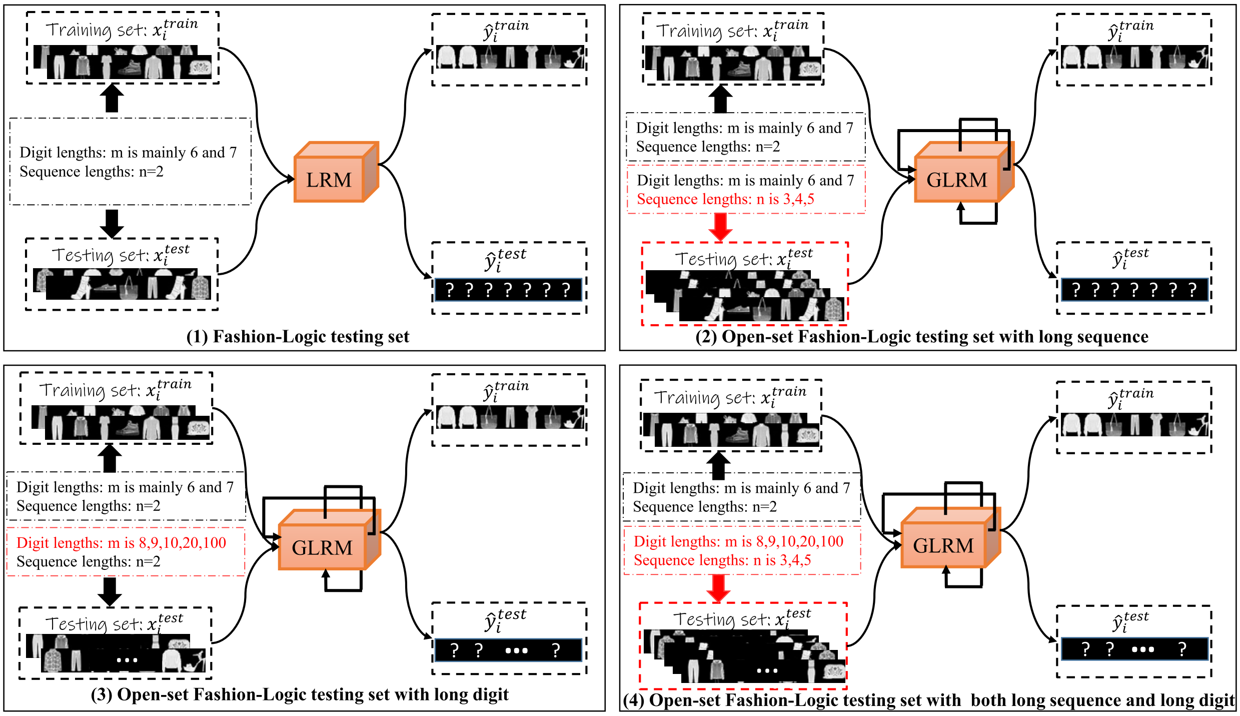

Training set: $x_i^{train}$
Digit lengths: m is mainly 6 and 7
Sequence lengths: n=2
Testing set: $x_i^{test}$
LRM
$\hat{y}_i^{train}$
$\hat{y}_i^{test}$
? ? ? ? ? ? ?
(1) Fashion-Logic testing set
Training set: $x_i^{train}$
Digit lengths: m is mainly 6 and 7
Sequence lengths: n=2
Digit lengths: m is mainly 6 and 7
Sequence lengths: n is 3,4,5
Testing set: $x_i^{test}$
GLRM
$\hat{y}_i^{train}$
$\hat{y}_i^{test}$
? ? ? ? ? ? ?
(2) Open-set Fashion-Logic testing set with long sequence
Training set: $x_i^{train}$
Digit lengths: m is mainly 6 and 7
Sequence lengths: n=2
Digit lengths: m is 8,9,10,20,100
Sequence lengths: n=2
Testing set: $x_i^{test}$
GLRM
$\hat{y}_i^{train}$
$\hat{y}_i^{test}$
? ? ••• ?
(3) Open-set Fashion-Logic testing set with long digit
Training set: $x_i^{train}$
Digit lengths: m is mainly 6 and 7
Sequence lengths: n=2
Digit lengths: m is 8,9,10,20,100
Sequence lengths: n is 3,4,5
Testing set: $x_i^{test}$
GLRM
$\hat{y}_i^{train}$
$\hat{y}_i^{test}$
? ? ••• ?
(4) Open-set Fashion-Logic testing set with both long sequence and long digit

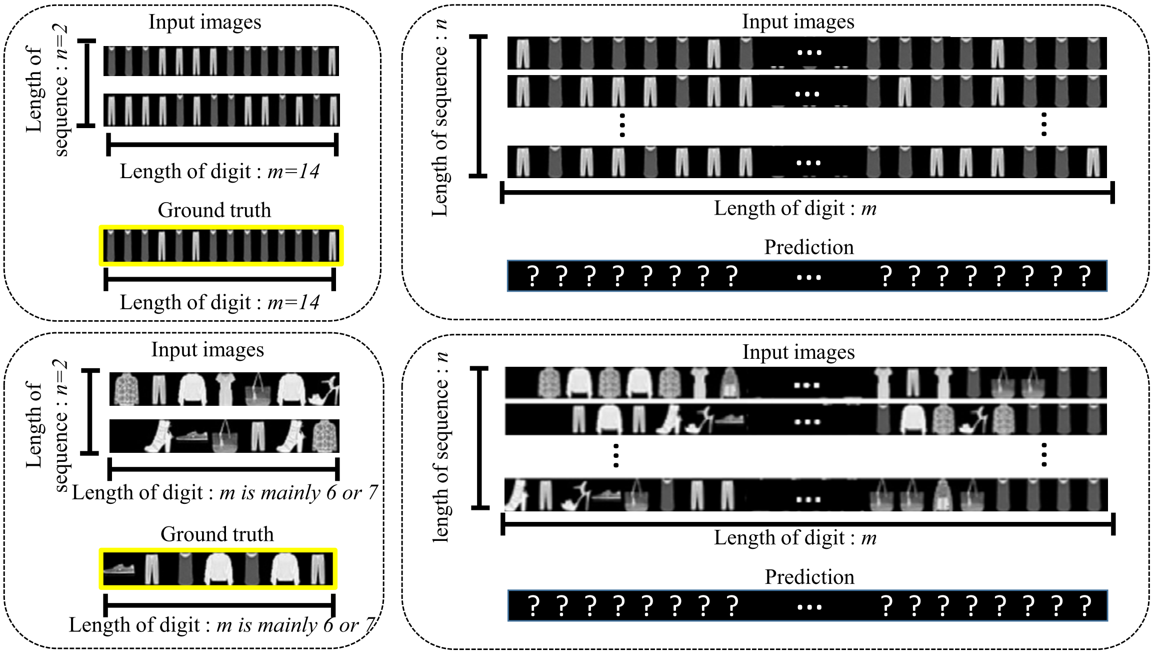
**Fig. 10.** The differences between the Fashion-Logic testing set, the open-set Fashion-Logic testing set with long sequence, the open-set Fashion-Logic testing set with long digit and the open-set Fashion-Logic testing set with both long sequence and long digit.

open-set Fashion-Logic testing set whose sequence length of examples is greater than 2 or digit length of examples is longer than 14 and 7 for bitwise logic and arithmetic logic, respectively, i.e., $m > 14$ for bitwise logic, and $m > 7$ for arithmetic logic. The configurations of experiments in this section are the same as that in Appendix, which is not described here again. Its effectiveness is validated with extensive experiments. The GLRM takes the open-set Fashion-Logic testing input set as input, granulates input images into subimages, reasons the logic embedded in each subimage and organizes these logic results. The experimental results are shown in Table 5, Tables 7-10.

### 5.1. Open-set Fashion-Logic data set and open-set Fashion-Logic task

Studies [54,55] have proposed the DBA and RBA data sets and made progress in the case of inconsistent data distribution, but there is still a lot of work need to be done. The logic relation hidden in the DBA and RBA is simple binary logic, and the task of [54,55] is to determine whether the equation embedded in the image is true or not. More complex decimal logic should be considered, and the task of directly giving the logic result of the images should be proposed.

To further explore the logic learning in the case of inconsistent data distribution, more challenging open-set Fashion-Logic data set is proposed. Specifically, for the digit length, the examples from the training set are generated by the distribution of digit lengths of 14 (mainly 6 and 7) on Bitwise And and Bitwise Or (Addition and Substraction). For the sequence length, the examples from the training set are generated by the distribution of sequence length of 2. In order to validate the effectiveness of the GLRM on the sequence dimension, for the examples from the testing set, the digit length is 14 (mainly 6 and 7) on Bitwise And and Bitwise Or (Addition and Substraction), the sequence length is set as {3,4,5}. In order to validate the effectiveness of the GLRM on the digit dimension, for the examples from the testing set, the sequence length is fixed as 2, the digit length is set as {15, 16, 17, 20, 100}({8, 9, 10, 20, 100}) on Bitwise And and Bitwise Or (Addition and Substraction). In order to validate the effectiveness of the GLRM on both digit and sequence dimensions, for the examples from the testing set, the digit length is set as {15, 16, 17, 20, 100} ({8, 9, 10, 20, 100}) on Bitwise And and Bitwise Or (Addition and Substraction), the sequence length is set as {3,4,5}. In fact, the data distribution of the training set of the open-set Fashion-Logic and the Fashion-Logic is consistent. Therefore, we directly take the training set of the Fashion-Logic as the training set of the open-set Fashion-Logic. Fig. 11 is two examples on the open-set Fashion-Logic data set. Fig. 11 gives some examples on the open-set Fashion-Logic data set. The equations embedded in Fig. 11(a) mean "00011110000001 & 11110100110101 = 00010100000001" and "6123825 + 978196 = 7102021". Similarly, the equations embedded in Fig. 11(b) are a "bitwise and" equation and a "addition" equation, and their results need to be predicted through a network. Since only part of the examples is shown in Fig. 11(b), we will not discuss the specific values in the equations in Fig. 11(b). The open-set Fashion-Logic data set is markedly different from the DBA and RBA data sets. Compared with the DBA and RBA data sets, the open-set Fashion-Logic data set has the following challenges:



(a) Training examples (b) Testing examples

**Fig. 11.** Two examples on the open-set Fashion-Logic data set. The first row for the Bitwise And; the second row for Addition.

(1) The DBA and RBA contain binary logic, whereas the open-set Fashion-Logic contains more complex decimal logic.
(2) The DBA and RBA only contain binary addition, whereas the open-set Fashion-Logic contains four logic relations (decimal addition is one of them).
(3) The maximum digit length of the number (100) embedded in each example from the open-set Fashion-Logic is much larger than that (26) embedded in each example from the DBA and RBA.
(4) The sequence length of each example from the DBA and RBA is 2, whereas the sequence length of each example from the open-set Fashion-Logic is N (N>2).
(5) For DBA and RBA, it is necessary to judge whether the equation embedded in each example is right or not; however, for the open-set Fashion-Logic, the logic result of the formula embedded in each image needs to be given.

Furthermore, we propose the open-set Fashion-Logic task, which is very different from the Fashion-Logic task. The open-set Fashion-Logic task aims to mine the logic pattern hidden in the data set in the case of inconsistent data distribution. It is worth noting that the machine does not know the meanings of the figure symbols embedded in images and logic relations among images in advance. Specifically, a machine mines the logic pattern between the input and output images of training set, and then uses the reasoning method and the obtained logic pattern to analyse the testing set with long sequence and long digit. After the analysis, the machine predicts the logic results of the testing set, so as to solve the open-set Fashion-Logic task. The open-set Fashion-Logic task is very challenging and important for logic learning.

### *5.2. Experimental effectiveness of GLRM on the open-set Fashion-Logic testing set with long sequence*

In what follows, we explore the effectiveness of GLRM on the open-set Fashion-Logic testing set with long sequence, i.e., $n > 2$. In this experiment, the digit length $m$ of testing set is the same as that of the training set, and the sequence length $n$ of testing set is set to be $\{3, 4, 5\}$.

The experimental results are shown in Fig. 12 and Table 5. Fig. 12 shows the visual effects with the GLRMs on the open-set Fashion-Logic testing set with long sequence. From Table 5 and Fig. 12, one can get three observations.

(1) For bitwise logic, all GLRMs achieve the accuracies of 100%. For arithmetic logic, the GLRM with ResNets obtains very high accuracy and is able to keep good performance as the sequence length $n$ increases. This means that GLRMs have good generalization on the testing set that are different from the training set.
(2) The accuracies of GLRMs with CNN-LSTM, MLP, CNN-MLP and Autoencoder drop as the sequence length $n$ increases. For example, the accuracy of GLRM with CNN-MLP drops by 97.90%-67.38%=30.52% when $n$ changes from 3 to 5 on Subtraction. The reason is that the middle input $z_i^{(j)}$ (shown in Fig. 12) has the small amount noisy (such as blur, even wrong figure symbols). Specifically, $z_i^{(j)}$ is the output of the current LRM, which is slightly different from the real

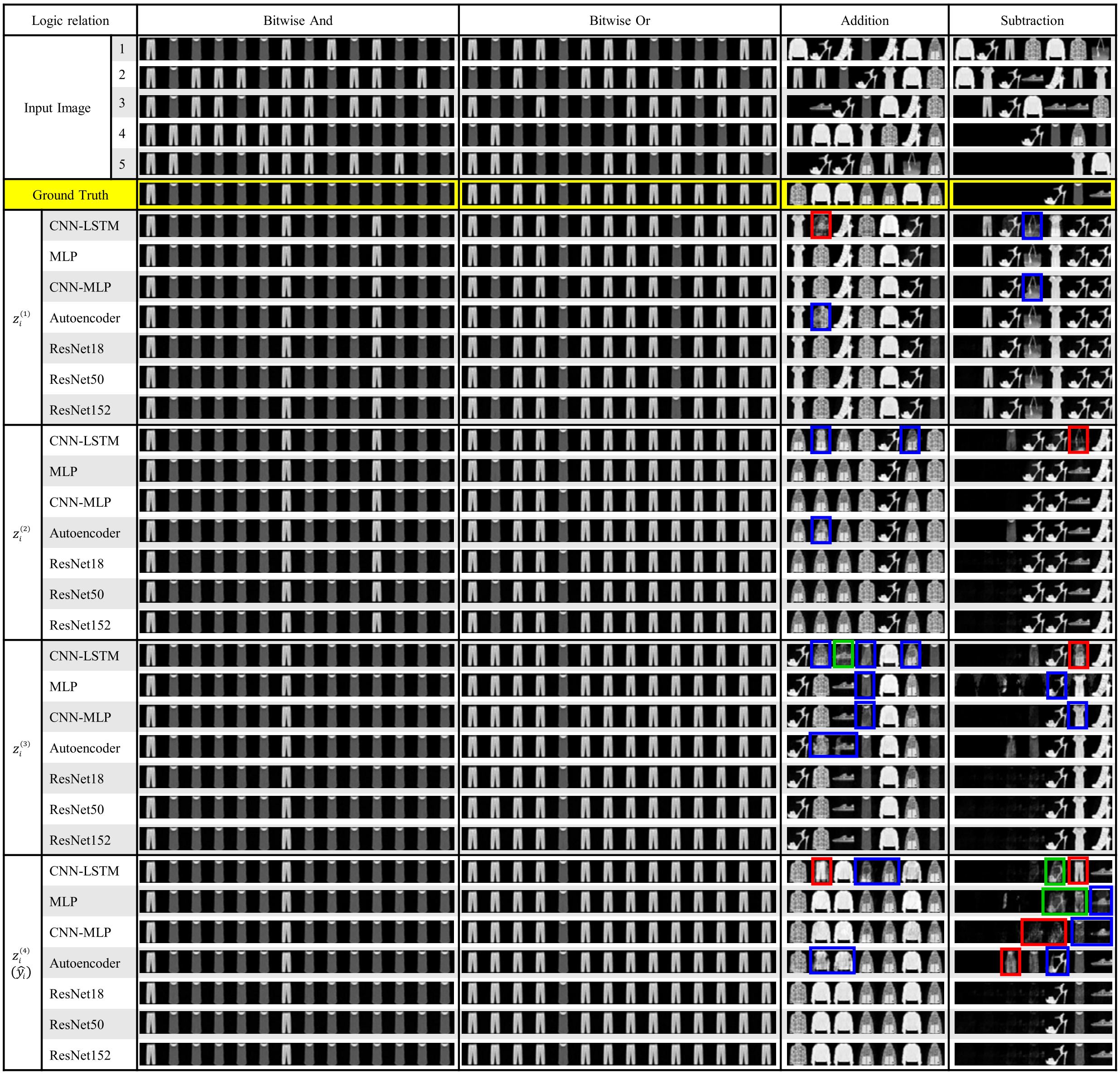


**Fig. 12.** The visual effects of the GLRM on the open-set Fashion-Logic testing set with long sequence. The blurry but right figure symbols predicted are marked with blue boxes, blurry and wrong those are marked with green boxes, and wrong those are marked with red boxes.

result. At the same time, $z_i^{(j)}$ serves as the input to the next LRM. Hence, such difference will accumulate continuously, resulting in a large deviation in the final output.

(3) Obviously, the LRM with the higher reasoning performance can more possibly reason true and clear $z_i^{(j)}$. This means that a LRM plays an important role in GLRM.

Based on these observations, it is concluded that GLRM with high-performance LRM is able to reason the hidden logic pattern from the long sequence visual data.

The distributions of the testing set are different from those of the training set when $n$ is greater than 2, which is shown in Table 6. For example, the row 1 on Addition [configuration: Train/$n = 2$] and row 2∼5 on Addition [configuration: Test/$n = 2$, Test/$n = 3$, Test/$n = 4$ and Test/$n = 5$] represent the data distribution of the training set and testing set, respectively. For $x_i^{(1)}$, $x_i^{(2)}$ and $y_i$ on Addition [configuration: Train/$n = 2$ and Test/$n = 2$], each data distribution looks like a right triangle. This means the data distribution of training set and testing set is consistent when the sequence length $n = 2$. However, the data distributions of $(x_i^{(1)} + x_i^{(2)})$ and $y_i$ on Addition [configuration: Test/$n = 3$] look just like a normal distribution and a

**Table 5**
The testing accuracies on the open-set Fashion-Logic data set with long sequence (m = 14 for Bitwise And and Bitwise Or, m is mainly 6 and 7 for Addition and Substraction).

| Logic relation | $n$ | CNN-LSTM | MLP | CNN-MLP | Autoencoder | ResNet18 | ResNet50 | ResNet152 |
|---|---|---|---|---|---|---|---|---|
| Bitwise And | 3 | 100% | 100% | 100% | 100% | 100% | 100% | 100% |
| | 4 | 100% | 100% | 100% | 100% | 100% | 100% | 100% |
| | 5 | 100% | 100% | 100% | 100% | 100% | 100% | 100% |
| Bitwise Or | 3 | 100% | 100% | 100% | 100% | 100% | 100% | 100% |
| | 4 | 100% | 100% | 100% | 100% | 100% | 100% | 100% |
| | 5 | 100% | 100% | 100% | 100% | 100% | 100% | 100% |
| Addition | 3 | 27.68% | 94.92% | 99.04% | 92.09% | 99.93% | 99.96% | 100% |
| | 4 | 14.31% | 89.73% | 93.23% | 84.24% | 99.94% | 99.96% | 100% |
| | 5 | 8.40% | 84.89% | 79.06% | 74.09% | 99.93% | 99.96% | 100% |
| Subtraction | 3 | 27.24% | 91.30% | 97.90% | 93.59% | 99.91% | 100% | 100% |
| | 4 | 16.16% | 82.65% | 90.01% | 88.71% | 99.84% | 99.99% | 100% |
| | 5 | 10.12% | 72.47% | 67.38% | 84.63% | 99.78% | 99.95% | 99.98% |

long-tailed distribution. This means the data distribution of training set and testing set is inconsistent when the sequence length $n = 3$.

### 5.3. Experimental effectiveness of GLRM on the open-set Fashion-Logic testing set with long digit

In this subsection, we explore the effectiveness of GLRM on the open-set Fashion-Logic testing set with long digit, i.e., $m > 7/14$. In this experiment, the sequence length $n$ of testing set is fixed as 2, and the digit length $m$ of testing set is set as $\{15, 16, 17, 20, 100\}/\{8, 9, 10, 20, 100\}$.

The experimental results are shown in Table 7. From Table 7, one can get three observations.

(1) For bitwise logic, all GLRMs achieve accuracies of 100%. For arithmetic logic, the GLRMs with ResNets obtain very high accuracies. Also, they still can keep good performances as the digit length $m$ increases. For example, GLRM with ResNet152 achieve 100% and 99.50% accuracies when the digit length $m$ increases to 100 which is longer 92 than digit length of the training examples on Addition and Subtraction. This means that GLRMs have good generalizations on the testing set whose digit length is longer than that of the training set.
(2) The accuracies of GLRMs with CNN-LSTM, MLP, CNN-MLP and Autoencoder faster decrease as the digit length $m$ increases. For example, the accuracy of GLRM with MLP drops by 98.00%-34.80%=63.20% when $m$ changes from 8 to 100 on Subtraction, which shows that long digit indeed brings a very big challenge for logic learning task. The challenge is that the hidden logic between $x_i^1$ and $x_i^2$ reasoned by GLRM is right only if all logic patterns between $\lceil \frac{m-k}{s} \rceil + 1$ subimage pairs $(x_i^{(1,l)}, x_i^{(2,l)})$ are right.
(3) The GLRMs have consistent performances with LRMs by observing Tables 1 and 7, i.e., a GLRM involving the LRM with the higher reasoning performance also can obtain better performance.

Based on these observations, it is further proved that GLRM with high-performance LRM is effective to reason the hidden logic pattern from the longer digit visual data.

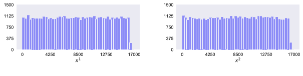
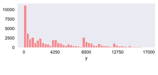
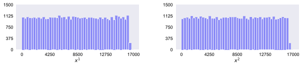
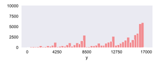
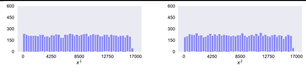
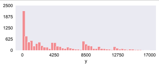
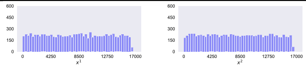
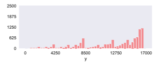
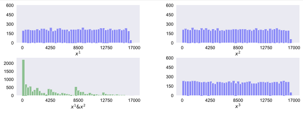
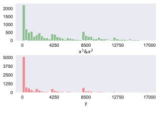
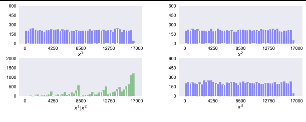
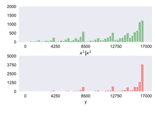
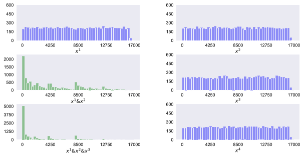
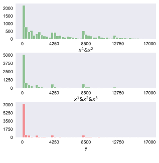
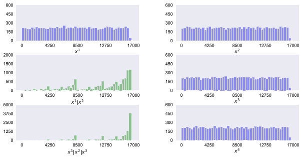
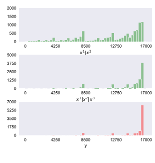
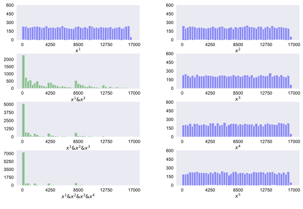
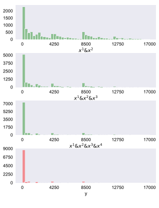
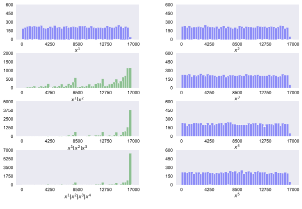
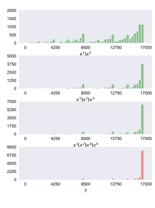
### 5.4. Experimental effectiveness of GLRM on the open-set Fashion-Logic testing set with both long sequence and long digit

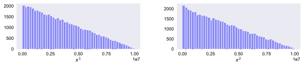
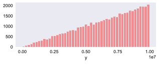
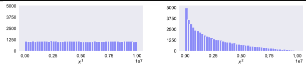
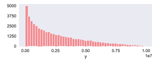
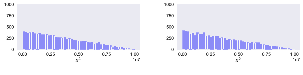
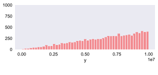
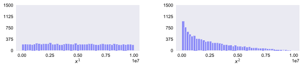
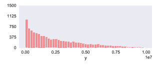
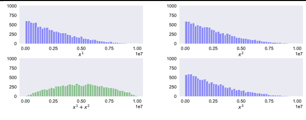
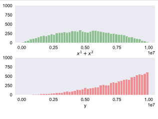
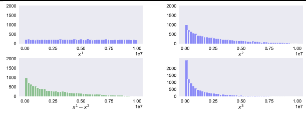
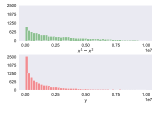
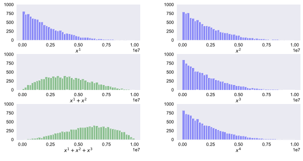
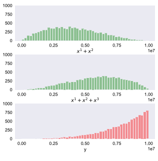
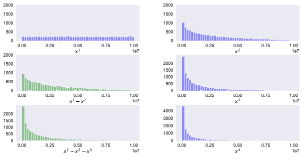
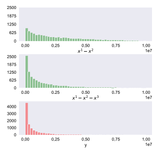
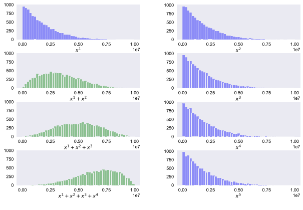
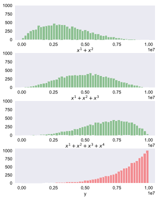
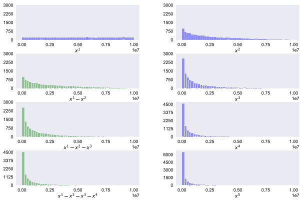
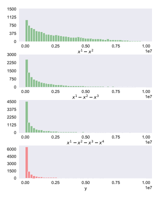
In this subsection, we explore the effectiveness of GLRM on the open-set Fashion-Logic testing set with both long sequence and long digit, i.e., $n > 2$ and $m > 7/14$. In this experiment, when the value of one parameter varies, the value of the other parameter is fixed. Here, the sequence length $n$ of testing set is set as $\{3, 4, 5\}$, and the digit length $m$ of testing set is set as $\{15, 16, 17, 20, 100\}/\{8, 9, 10, 20, 100\}$. Table 8 reports the detailed experimental results of each comparing approach in terms of different parameter configurations.

Note that the all GLRMs achieve the accuracies of 100% on bitwise logic. Hence, we do not show these results in Table 8. Based on the reported experimental results, for arithmetic logic, the following observations of the comparative studies can be made:

(1) Compared to the results shown in Table 3 that even though $n$ is set as 2 and $m$ is set as 8, accuracies of all LRMs except Autoencoder are almost 0, GLRMs obtain highly competitive generalization performances on the open-set Fashion-Logic testing set. For example, all GLRMs except for GLRM with CNN-LSTM achieve impressive performance that is not lower than 64% when $n$ is set as 3 and $m$ is set as 20. These results support our motivation that granular technique is helpful for logic learning by parting the complex visual logic into easier parts.
(2) Moreover, the accuracies of most GLRMs drop considerably as $m$ or $n$ increases. This coincides with our intuition that the reasoning difficulty indeed increases as the two parameters increase. Compared to Addition, accuracy for Substraction

**Table 6**
The data distributions on the open-set Fashion-Logic data set (m = 14 for Bitwise And and Bitwise Or, m is mainly 6 and 7 for Addition and Substraction).

| Conf. | Bitwise And | | Bitwise Or | |
|---|---|---|---|---|
| | Input images | Output image | Input images | Output image |
| Train $n = 2$ |  |  |  |  |
| Test $n = 2$ |  |  |  |  |
| Test $n = 3$ |  |  |  |  |
| Test $n = 4$ |  |  |  |  |
| Test $n = 5$ |  |  |  |  |

| Conf. | Addition | | Subtraction | |
|---|---|---|---|---|
| | Input images | Output image | Input images | Output image |
| Train $n = 2$ |  |  |  |  |
| Test $n = 2$ |  |  |  |  |
| Test $n = 3$ |  |  |  |  |
| Test $n = 4$ |  |  |  |  |
| Test $n = 5$ |  |  |  |  |

drops more when $n = 5$ and $m = 100$ (this condition is very challenging). For example, the accuracy of GLRM with ResNet152 drops by 6.10% compared to that in the setting $n = 5$ and $m = 8$. The phenomenon coincides with the visual effect results shown in Fig. 12 that GLRMs reason more wrong figure symbols marked with red and green boxes when

**Table 7**
The testing accuracies on the open-set Fashion-Logic data set with long digit ($n = 2$).

| Logic relation | $m$ | CNN-LSTM | MLP | CNN-MLP | Autoencoder | ResNet18 | ResNet50 | ResNet152 |
|---|---|---|---|---|---|---|---|---|
| Bitwise And | 15 | 100% | 100% | 100% | 100% | 100% | 100% | 100% |
| | 16 | 100% | 100% | 100% | 100% | 100% | 100% | 100% |
| | 17 | 100% | 100% | 100% | 100% | 100% | 100% | 100% |
| | 20 | 100% | 100% | 100% | 100% | 100% | 100% | 100% |
| | 100 | 100% | 100% | 100% | 100% | 100% | 100% | 100% |
| Bitwise Or | 15 | 100% | 100% | 100% | 100% | 100% | 100% | 100% |
| | 16 | 100% | 100% | 100% | 100% | 100% | 100% | 100% |
| | 17 | 100% | 100% | 100% | 100% | 100% | 100% | 100% |
| | 20 | 100% | 100% | 100% | 100% | 100% | 100% | 100% |
| | 100 | 100% | 100% | 100% | 100% | 100% | 100% | 100% |
| Addition | 8 | 50.00% | 98.70% | 99.80% | 97.10% | 99.90% | 99.90% | 100% |
| | 9 | 44.70% | 97.70% | 99.80% | 95.60% | 100% | 100% | 100% |
| | 10 | 36.50% | 97.70% | 99.90% | 95.30% | 100% | 100% | 100% |
| | 20 | 8.90% | 96.40% | 99.10% | 86.80% | 99.90% | 100% | 100% |
| | 100 | 0.00% | 79.90% | 91.50% | 51.60% | 99.00% | 99.70% | 100% |
| Subtraction | 8 | 44.40% | 98.00% | 99.70% | 96.80% | 100% | 100% | 100% |
| | 9 | 41.90% | 96.50% | 99.20% | 96.20% | 99.90% | 100% | 100% |
| | 10 | 36.50% | 97.20% | 99.00% | 96.20% | 100% | 100% | 100% |
| | 20 | 10.80% | 90.60% | 97.20% | 89.30% | 99.90% | 99.90% | 99.90% |
| | 100 | 0.00% | 34.80% | 74.00% | 43.60% | 98.60% | 97.70% | 99.50% |

**Table 8**
The testing accuracies on the open-set Fashion-Logic data set with both long sequence and long digit.

| Logic relation | $n$ | $m$ | CNN-LSTM | MLP | CNN-MLP | Autoencoder | ResNet18 | ResNet50 | ResNet152 |
|---|---|---|---|---|---|---|---|---|---|
| Addition | 3 | 8 | 21.30% | 93.30% | 98.30% | 88.40% | 99.90% | 99.90% | 100% |
| | | 9 | 14.40% | 93.60% | 96.70% | 87.30% | 99.90% | 100% | 100% |
| | | 10 | 12.00% | 93.00% | 96.70% | 84.10% | 99.90% | 100% | 100% |
| | | 20 | 0.80% | 89.20% | 89.70% | 69.60% | 99.70% | 99.80% | 100% |
| | | 100 | 0.00% | 47.70% | 40.00% | 8.90% | 97.40% | 99.00% | 99.90% |
| | 4 | 8 | 8.30% | 89.00% | 78.50% | 80.00% | 99.70% | 99.80% | 99.90% |
| | | 9 | 6.30% | 90.10% | 71.40% | 76.90% | 99.80% | 99.80% | 100% |
| | | 10 | 3.40% | 87.10% | 69.00% | 73.70% | 99.90% | 100% | 100% |
| | | 20 | 0.10% | 78.70% | 34.40% | 48.30% | 99.50% | 99.80% | 100% |
| | | 100 | 0.00% | 25.20% | 0.00% | 1.00% | 95.80% | 98.00% | 99.80% |
| | 5 | 8 | 4.00% | 83.60% | 45.30% | 69.20% | 99.80% | 99.80% | 100% |
| | | 9 | 3.10% | 81.40% | 34.60% | 63.70% | 99.90% | 99.90% | 100% |
| | | 10 | 0.80% | 82.30% | 27.40% | 61.70% | 99.90% | 100% | 100% |
| | | 20 | 0.00% | 68.80% | 3.00% | 29.10% | 99.00% | 99.80% | 100% |
| | | 100 | 0.00% | 12.70% | 0.00% | 0.00% | 93.10% | 98.70% | 99.90% |
| Subtraction | 3 | 8 | 17.80% | 88.30% | 95.10% | 91.30% | 99.70% | 100% | 99.90% |
| | | 9 | 16.50% | 84.90% | 95.00% | 92.00% | 100% | 100% | 100% |
| | | 10 | 12.30% | 84.60% | 94.40% | 88.10% | 99.80% | 99.90% | 100% |
| | | 20 | 0.80% | 64.00% | 85.00% | 75.20% | 99.10% | 99.30% | 99.80% |
| | | 100 | 0.00% | 0.20% | 16.50% | 11.90% | 90.90% | 91.70% | 97.50% |
| | 4 | 8 | 11.10% | 78.80% | 71.50% | 87.10% | 99.90% | 99.90% | 100% |
| | | 9 | 7.90% | 71.50% | 65.00% | 83.30% | 99.60% | 99.70% | 99.80% |
| | | 10 | 5.50% | 69.60% | 57.70% | 81.50% | 99.40% | 99.50% | 99.70% |
| | | 20 | 0.10% | 38.90% | 25.00% | 59.20% | 97.40% | 98.00% | 99.30% |
| | | 100 | 0.00% | 0.00% | 0.00% | 3.00% | 84.20% | 85.80% | 95.60% |
| | 5 | 8 | 5.00% | 68.30% | 36.10% | 77.10% | 99.10% | 99.50% | 99.90% |
| | | 9 | 2.20% | 60.70% | 22.90% | 76.00% | 99.20% | 99.50% | 99.90% |
| | | 10 | 2.10% | 53.90% | 17.80% | 71.00% | 98.50% | 98.70% | 99.40% |
| | | 20 | 0.00% | 20.70% | 1.20% | 44.90% | 96.20% | 97.30% | 98.70% |
| | | 100 | 0.00% | 0.00% | 0.00% | 0.40% | 75.80% | 81.10% | 93.80% |

$n = 5$. One reason is that the middle input $z_i^{j,l}$ has the small amount noisy (such as blur) because itself is also the output of the previous LRM. Hence, a more robust LRM for noisy needs to be further explored in the future by taking into account data uncertainty in logic learning.

(3) Also, it is impressive that GLRM with ResNet152 achieves 99.90% accuracy on Addition even though $n$ and $m$ increase to 5 and 100, respectively. However, the accuracies of GLRMs with CNN-LSTM, MLP, CNN-MLP and Autoencoder drop to

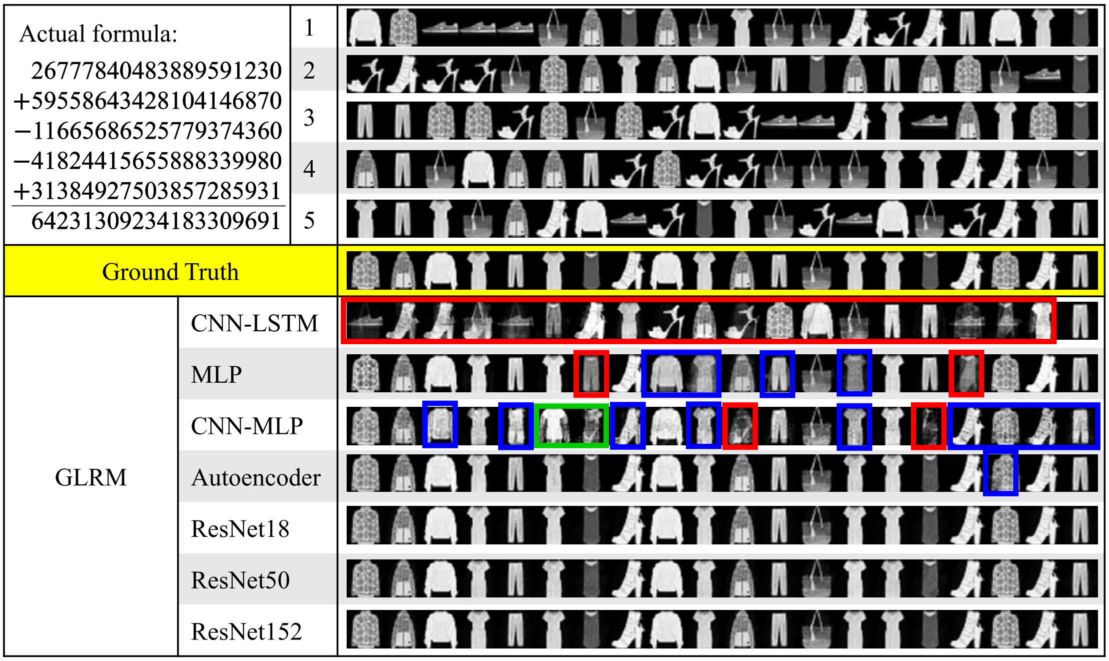


**Fig. 13.** The visual effects on a mixture logic of addition and subtraction. The blurry but right figure symbols predicted are marked with blue box, blurry and wrong those are marked with green box, and wrong those are marked with red box.

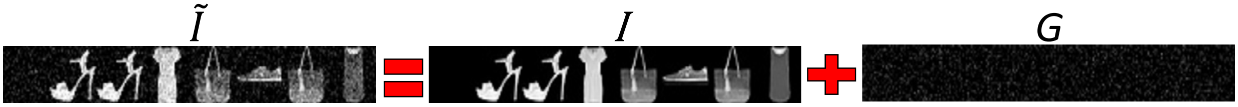


**Fig. 14.** The example of noisy data.

0.00%, 12.70%, 0.00% and 0.00% on Addition, respectively. This further indicates a base LRM with high performance plays a very important role in GLRM.

Overall, all GLRMs benefit from granular technique. Furthermore, we extend the GLRMs on a mixture open-set Fashion-Logic task with addition and subtraction. Fig. 13 gives an illustrative example to show the visual effects of GLRMs with different LRMs on a mixed logic. The result shows that GLRMs with ResNets still reasons the right logic pattern that is hidden in the visual data, and the reasoned figure symbols are very clear. We believe that this additional result shows the generality and robustness of the GLRM.

### 5.5. Experimental effectiveness of GLRM on the noisy data

In this section, the effectiveness of GLRM is validated on noisy data. According to the above analyses, one knows that in the open-set Fashion-Logic testing set, (1) the arithmetic logic is more difficult than the bitwise logic; (2) the arithmetic logic with sequence length $n = 5$ is more difficult than that with sequence length $n = \{2, 3, 4\}$. Therefore, taking the Addition and Subtraction with sequence length $n = 5$ as an example, the effectiveness research of GLRM is carried out by adding gaussian noises with different degrees to data. Specifically, the gaussian noises ($\mu$=0, $\sigma$={10, 20, 30}) are added to all input and output images of Addition and Subtraction with sequence length $n = 5$ and digit length $m = \{8, 9, 10, 20, 100\}$, which is formalized as follows:

$$\tilde{I}(x, y) = I(x, y) + G(\mu, \sigma), \tag{11}$$

where $I$ denotes original image, $I(x, y) \in (0, 255)$ denotes the value of each pixel of original image. $G(\mu, \sigma)$ denotes gaussian noisy signal with mean value $\mu$ and standard deviation $\sigma$, in this paper, $\mu$ = 0, $\sigma$={10, 20, 30}, $\tilde{I}$ denotes noisy image, $\tilde{I}(x, y)$ denotes the value of each pixel of noisy image. If $\tilde{I}(x, y) \leq 0$, then $\tilde{I}(x, y) = 0$; if $\tilde{I}(x, y) \geq 255$, then $\tilde{I}(x, y) = 255$. An example is shown in Fig. 14.

**Table 9**
The testing accuracies of GLRMs on the noisy Addition and Subtraction (sequence length $n = 5$).

| Logic relation | $\sigma$ | $m$ | CNN-LSTM | MLP | CNN-MLP | Autoencoder | ResNet18 | ResNet50 | ResNet152 |
|---|---|---|---|---|---|---|---|---|---|
| Addition | 10 | 8 | 3.30% | 83.20% | 3.60% | 65.80% | 99.70% | 99.80% | 100% |
| | | 9 | 1.90% | 80.60% | 1.50% | 61.20% | 99.80% | 99.90% | 100% |
| | | 10 | 0.70% | 82.60% | 0.60% | 60.10% | 99.90% | 100% | 100% |
| | | 20 | 0.00% | 65.70% | 0.00% | 26.70% | 99.10% | 99.80% | 100% |
| | | 100 | 0.00% | 9.50% | 0.00% | 0.00% | 92.60% | 98.30% | 99.90% |
| | 20 | 8 | 0.90% | 76.90% | 0.00% | 56.60% | 99.70% | 99.70% | 100% |
| | | 9 | 0.20% | 73.50% | 0.00% | 49.30% | 99.90% | 99.80% | 100% |
| | | 10 | 0.10% | 75.70% | 0.00% | 48.10% | 99.90% | 99.80% | 100% |
| | | 20 | 0.00% | 56.10% | 0.00% | 17.40% | 99.00% | 99.30% | 99.90% |
| | | 100 | 0.00% | 3.00% | 0.00% | 0.00% | 91.40% | 96.90% | 99.90% |
| | 30 | 8 | 0.00% | 62.50% | 0.00% | 41.50% | 98.30% | 97.50% | 99.70% |
| | | 9 | 0.00% | 59.30% | 0.00% | 36.00% | 98.50% | 96.70% | 99.30% |
| | | 10 | 0.00% | 61.40% | 0.00% | 30.50% | 98.70% | 95.10% | 99.50% |
| | | 20 | 0.00% | 32.30% | 0.00% | 7.10% | 95.50% | 89.40% | 99.10% |
| | | 100 | 0.00% | 0.40% | 0.00% | 0.00% | 73.20% | 62.50% | 95.40% |
| Subtraction | 10 | 8 | 4.10% | 68.80% | 0.30% | 77.80% | 99.00% | 99.50% | 99.90% |
| | | 9 | 2.40% | 62.50% | 0.10% | 75.80% | 99.20% | 99.50% | 99.90% |
| | | 10 | 1.10% | 54.10% | 0.10% | 70.50% | 98.30% | 98.90% | 99.40% |
| | | 20 | 0.00% | 20.70% | 0.00% | 42.90% | 96.10% | 97.40% | 98.70% |
| | | 100 | 0.00% | 0.00% | 0.00% | 0.00% | 75.70% | 80.50% | 93.20% |
| | 20 | 8 | 1.50% | 60.40% | 0.00% | 74.40% | 96.70% | 99.20% | 99.90% |
| | | 9 | 0.30% | 52.30% | 0.00% | 71.10% | 96.80% | 99.20% | 99.90% |
| | | 10 | 0.30% | 47.00% | 0.00% | 63.70% | 95.90% | 98.30% | 99.30% |
| | | 20 | 0.00% | 15.40% | 0.00% | 34.50% | 92.00% | 96.40% | 98.70% |
| | | 100 | 0.00% | 0.00% | 0.00% | 0.10% | 55.70% | 75.30% | 92.10% |
| | 30 | 8 | 0.00% | 43.00% | 0.00% | 65.00% | 79.40% | 90.10% | 99.10% |
| | | 9 | 0.00% | 38.00% | 0.00% | 63.50% | 76.10% | 88.90% | 98.70% |
| | | 10 | 0.00% | 32.40% | 0.00% | 55.10% | 74.30% | 84.50% | 97.80% |
| | | 20 | 0.00% | 6.60% | 0.00% | 22.50% | 57.40% | 71.60% | 94.80% |
| | | 100 | 0.00% | 0.00% | 0.00% | 0.00% | 3.40% | 17.40% | 76.00% |

The GLRMs are used directly without being retrained, and the experimental results are shown in Table 9. From Table 9, one can get three observations.

(1) After adding the slight noise $G(0, 10)$, the performances of all GLRMs except for the GLRM with CNN-MLP are little or no affected. When the noise increases to $G(0, 20)$, for Addition, the performances of the GLRMs with ResNets are little or no affected; for Subtraction, the performances of the GLRM with ResNet50 and the GLRM with ResNet152 are little or no affected. This indicates that most GLRM has certain robustness and can tolerate noise to a certain extent.
(2) When the noise continues to increase to $G(0, 30)$, for Addition, it is impressive that the GLRM with ResNet152 still reaches 95.4% accuracy when digit length $m$ = 100, and its performance is little affected. However, for Subtraction, the performances of all GLRMs are reduced. This indicates that Subtraction is more difficult than Addition. According to the change trend of performance with the noisy degree, when the noise reaches a high degree, the performances of all GLRMs tend to decrease.
(3) From the perspective of GLRMs, the performance of GLRM with ResNet152 declines slowly with the increase of noise, which indicates that GLRM with ResNet152 has the strongest robustness. The performances of GLRMs with other LRMs decline rapidly, especially the GLRM with CNN-MLP, whose accuracies are all 0.00% when the noises are $G(0, 20)$ and $G(0, 30)$. This further indicates a base LRM with high performance plays a very important role in GLRM.

On the one hand, the results on Table 9 show that most GLRMs are robust and can tolerate noise to a certain extent; on the other hand, as the noise continues to increase, the performances of GLRMs are bound to decline. So we retrain GLRMs using Subtraction with noise $G(0, 30)$ and observe whether the GLRMs can cope with noisy data. The experimental results are shown in Table 10.

From Table 10, one can see that after GLRMs are retrained on noisy data, they perform as well as or even better than on original data. For example, the accuracy of GLRM with CNN-MLP that is trained on the noisy data increases from 0.00% to 90.00% ($m$= 8) compared to that without being retrained. This indicates that the GLRM with CNN-MLP trained on the noisy data is more robust and the noisy data can promote the effectiveness of GLRM to a certain extent.

Fig. 15 shows the visual effects of GLRMs on the original data and noisy data ($\sigma$= 30), where Figs. 15(a) and 15(b) show the visual effects of GLRMs on the Addition and the Subtraction, respectively. From Fig. 15, one can get three observations.

**Table 10**
The testing accuracies of GLRMs after being retrained on the noisy Subtraction (sequence length $n = 5$).

| Logic relation | $\sigma$ | $m$ | CNN-LSTM | MLP | CNN-MLP | Autoencoder | ResNet18 | ResNet50 | ResNet152 |
|---|---|---|---|---|---|---|---|---|---|
| Subtraction (after retraining) | 30 | 8 | 11.90% | 68.40% | 90.00% | 73.30% | 99.60% | 99.70% | 100% |
| | | 9 | 6.10% | 59.50% | 87.40% | 71.00% | 99.60% | 99.50% | 99.80% |
| | | 10 | 4.00% | 56.10% | 85.10% | 65.30% | 98.70% | 98.70% | 99.30% |
| | | 20 | 0.00% | 26.60% | 68.20% | 36.60% | 96.80% | 96.80% | 98.70% |
| | | 100 | 0.00% | 0.00% | 1.20% | 0.10% | 81.60% | 80.70% | 93.80% |

(1) The ground truth images are noisy, while the predicted images obtained through GLRMs are clean, which indicates that GLRM does learn the logic relation hidden in the image without learning noise. This validates that GLRM is indeed an effective logic learning method.

(2) Fig. 15(a) is an example of 8-digit addition equation “5538780 + 8038946 + 4151711 + 20334198 + 57246437 = 95310072”, the left and right are the visual effects of GLRMs on original data and noisy data, respectively. As can be seen from the figure, the GLRM with MLP and the GLRMs with Resnets are robust to noisy data, and their visual effects on noisy data are not different from those on original data. GLRM with Autoencoder is slightly affected by noisy data, and its visual effect slightly decreases. The GLRM with CNN-LSTM and the GLRM with CNN-MLP are greatly affected by noisy data, and their visual effects decrease obviously.

(3) Fig. 15(b) is an example of 8-digit subtraction equation “97808077 - 11645637 - 72655298 - 3499757 - 3483402 = 6523983”. The left and middle are the visual effects of GLRMs on original data and noisy data, respectively. As can be seen from the figure, all GLRMs are affected by noisy data, with a certain degree of visual deterioration. After the GLRMs are retrained using noisy data, their visual effects on noisy data (the right of Fig. 15(b)) are better than those on original data. This further visually demonstrates that the GLRMs after being retrained can handle noisy data, and the noisy data in turn enhance the robustness of GLRMs.

In summary, the GLRMs work well on the original data and slightly noisy data. After the GLRMs are retrained, they still perform well on high-noise data, even better than on original data. These results clearly validate the effectiveness of multigranulation strategy in mining hidden logic pattern from visual logic examples.

## 6. Conclusion

In this article, we focus on the problem of mining logic pattern directly from data in the case of inconsistent data distribution. Under the traditional machine learning paradigm, the LRMs can perform logic pattern mining task well. To further test the generalization performance of these methods, we generate the simple open-set Fashion-Logic data set with inconsistent data distribution. However, when the traditional machine learning paradigm is no longer satisfied, the performance of these methods has a great decline or even failure. In order to address the open-set logic reasoning task, the GLRM has been proposed with the inspiration of multigranulation studies. Its effectiveness has been demonstrated on the proposed open-set Fashion-Logic data set.

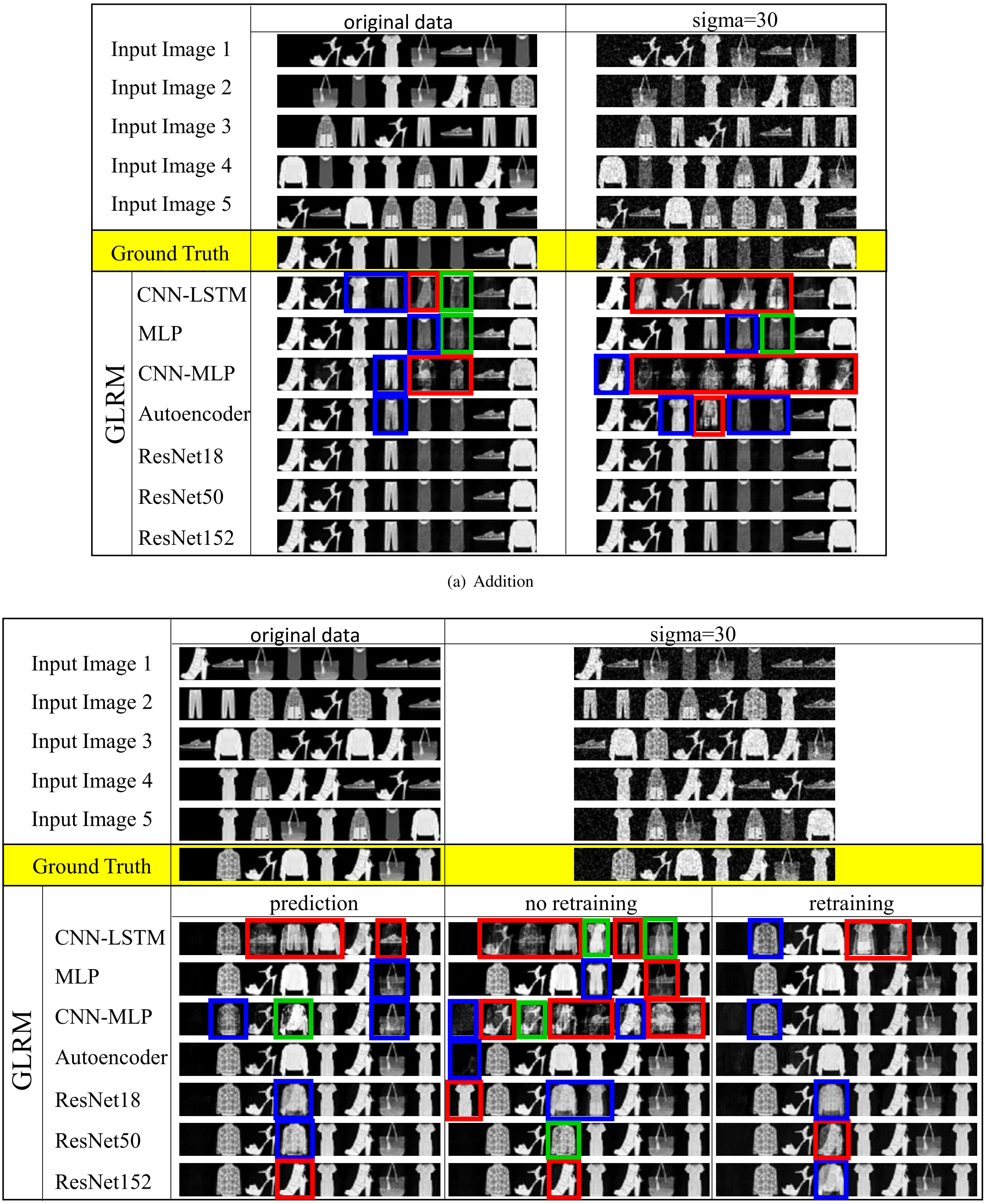
In the future, there are many interesting topics in logic learning. For example, the same logic may be hidden in different symbol systems, and hence it is desirable to develop a symbol-system-free logic reasoning machine. Moreover, a base LRM plays an important role in GLRM, and hence it is desirable to develop a more robust LRM for noisy (such as blur) by modeling data uncertainty in logic learning.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

This work was supported by National Key Research and Development Program of China (Nos. 2021ZD0112400, 2020AAA0106100), the Key Program of the National Natural Science Foundation of China (Nos. 62136005, 61906115, 62106132, 61906114), Key R&D Program (International Science and Technology Cooperation Project) of Shanxi Province, China (No. 201903D421003), the 1331 Engineering Project of Shanxi Province, Shanxi Province Natural Science Foundation for Youths (Nos. 20210302124549, 20210302124556, 201901D211171), Research Project Supported by Shanxi Scholarship Council of China (No. HGKY2019001), Scientific and Technologial Innovation Programs of Higher Education Institutions in Shanxi (No. 2020L0036), Natural Science Foundation of Shanxi Province, China (No. 20210302123455), and Program for the Young San Jin Scholars of Shanxi (No. 2016769).



(a) Addition

(b) Subtraction

**Fig. 15.** The visual effects of GLRMs on original data and noisy data ($\sigma$ = 30). The blurry but right figure symbols predicted are marked with blue box, blurry and wrong those are marked with green box, and wrong those are marked with red box.

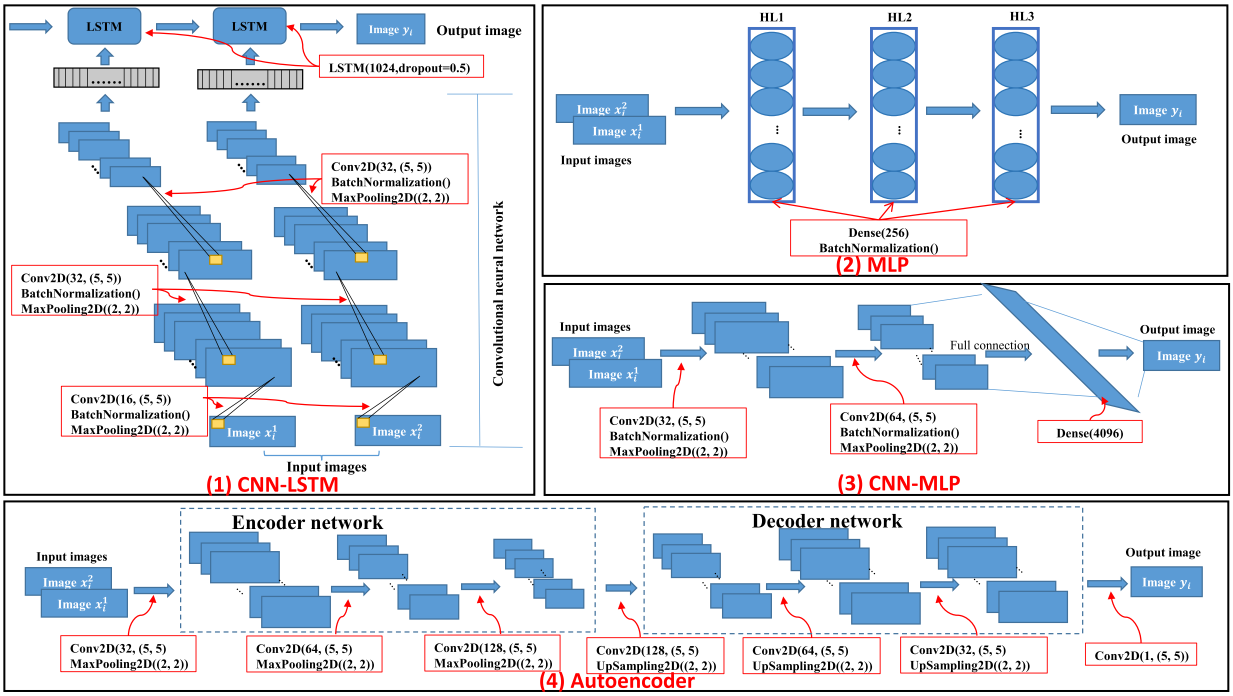


**Fig. A.1.** The structures of the LRMs.

## Appendix A

### A.1. The experimental setups

The models are implemented using Keras API (version: 2.1.6). The operating environment is Ubuntu 16.04.4, 512 GB DDR4 RDIMM, 2X 40-Core Intel(R) Xeon(R) CPU E5-2698 v4 @ 2.20GH and NVIDIA Tesla P100 with 16 GB GPU memory. The models are trained by optimising a mean square error loss. In order to avoid overfitting, early-stopping is used to select the hyper-parameters when the losses no longer decrease on the validation set. The batch size is set to be 32.

### A.2. The structures of the LRMs

The LRM is a logic reasoning machine shown in Fig. 4. Specifically, it can be any deep neural network. Without loss of generality, we randomly pick some typical deep neural networks like CNN-LSTM, MLP, CNN-MLP, Autoencoder and ResNets as LRMs. All LRMs used in this paper take two images as input and one image as output, respectively. The structures of the LRMs are shown in Fig. A.1 and they are described as follows.

**CNN-LSTM**: We design the CNN-LSTM based on LSTM [67]. We feed each image in sequence to a three-layer CNN, and then pass the outputs to the LSTM. We use the ReLU activation function, batch normalization and maxpooling layer after each convolution layer. The CNN-LSTM works best when the number of neurons of LSTM hidden layer is set to 1024 and the dropout is applied to the LSTM hidden state. At last, we use a fully-connected layer with sigmoid activation function after the output of the LSTM.

**MLP**: We use the MLP (multi-layer perceptron) as described in [37]. The MLP is made up of three fully-connected layers. The number of neurons of each fully-connected layer is 256. We use the ReLU activation function and batch normalization after each fully-connected layer. At last, we use another fully-connected layer with sigmoid activation function as the last layer of the MLP.

**CNN-MLP**: The CNN-MLP consists of a two-layer CNN and two fully-connected layers sequentially. We use the ReLU activation function, batch normalization and maxpooling layer after each convolution layer. We use the ReLU activation function and the sigmoid activation function after the first fully-connected layer and the last fully-connected layer, respectively.

**Autoencoder**: We implement a autoencoder using the idea of [68]. We use a three-layer CNN as the encoder and the decoder, respectively. For the encoder, we use the ReLU activation function and maxpooling layer after each convolution layer. For the decoder, we use the ReLU activation function and upsampling layer after each convolution layer. At last, we use a convolutional layer with sigmoid activation function after the output of the decoder.

**ResNet**: In this paper, we use ResNet-18, ResNet-50 and ResNet-152 as described in [69]. In order to meet the requirements of our data set, we replace the softmax activation function on the last layer with the sigmoid activation function.

## References

[1] S.M. Mousavi, S. Abdullah, S.T.A. Niaki, S. Banihashemi, An intelligent hybrid classification algorithm integrating fuzzy rule-based extraction and harmony search optimization: medical diagnosis applications, Knowl.-Based Syst. 220 (2021) 106943.

[2] H. Jiang, J. Zhan, B. Sun, J.C.R. Alcantud, An madm approach to covering-based variable precision fuzzy rough sets: an application to medical diagnosis, Int. J. Mach. Learn. Cybern. 11 (9) (2020) 2181–2207.

[3] B. Sun, X. Zhou, N. Lin, Diversified binary relation-based fuzzy multigranulation rough set over two universes and application to multiple attribute group decision making, Inf. Fusion 55 (2020) 91–104.

[4] R. Shang, J. Song, L. Jiao, Y. Li, Double feature selection algorithm based on low-rank sparse non-negative matrix factorization, Int. J. Mach. Learn. Cybern. (2020) 1–18.

[5] M.M. Javidi, Feature selection schema based on game theory and biology migration algorithm for regression problems, Int. J. Mach. Learn. Cybern. 12 (2) (2021) 303–342.

[6] C. Wang, Q. Hu, X. Wang, D. Chen, Y. Qian, Z. Dong, Feature selection based on neighborhood discrimination index, IEEE Trans. Neural Netw. Learn. Syst. 29 (7) (2017) 2986–2999.

[7] X. Liang, Q. Guo, Y. Qian, W. Ding, Q. Zhang, Evolutionary deep fusion method and its application in chemical structure recognition, IEEE Trans. Evol. Comput. 25 (5) (2021) 883–893, https://doi.org/10.1109/TEVC.2021.3064943.

[8] X. Liang, Y. Qian, Q. Guo, H. Cheng, J. Liang, AF: an association-based fusion method for multi-modal classification, IEEE Trans. Pattern Anal. Mach. Intell. (2021), https://doi.org/10.1109/TPAMI.2021.3125995.

[9] J. Li, C. Huang, J. Qi, Y. Qian, W. Liu, Three-way cognitive concept learning via multi-granularity, Inf. Sci. 378 (1) (2017) 244–263.

[10] X. Yang, S. Liang, H. Yu, S. Gao, Y. Qian, Pseudo-label neighborhood rough set: measures and attribute reductions, Int. J. Approx. Reason. 105 (2019) 112–129.

[11] Y. Yao, Three-way granular computing, rough sets, and formal concept analysis, Int. J. Approx. Reason. 116 (2020) 106–125.

[12] Q. Zhang, Y. Chen, J. Yang, G. Wang, Fuzzy entropy: a more comprehensible perspective for interval shadowed sets of fuzzy sets, IEEE Trans. Fuzzy Syst. 28 (11) (2019) 3008–3022.

[13] Y. Guo, E.C. Tsang, W. Xu, D. Chen, Local logical disjunction double-quantitative rough sets, Inf. Sci. 500 (2019) 87–112.

[14] Q. Hu, L. Zhang, S. An, D. Zhang, D. Yu, On robust fuzzy rough set models, IEEE Trans. Fuzzy Syst. 20 (4) (2011) 636–651.

[15] P. Zhang, T. Li, G. Wang, C. Luo, H. Chen, J. Zhang, D. Wang, Z. Yu, Multi-source information fusion based on rough set theory: a review, Inf. Fusion 68 (2021) 85–117.

[16] Y. She, X. He, H. Shi, Y. Qian, A multiple-valued logic approach for multigranulation rough set model, Int. J. Approx. Reason. 82 (2017) 270–284.

[17] W. Wu, Y. Qian, T. Li, S. Gu, On rule acquisition in incomplete multi-scale decision tables, Inf. Sci. 378 (2017) 282–302.

[18] Y. Qian, Y. Li, J. Liang, G. Lin, C. Dang, Fuzzy granular structure distance, IEEE Trans. Fuzzy Syst. 23 (6) (2015) 2245–2259.

[19] R. Yan, Y. Yu, D. Qiu, Emotion-enhanced classification based on fuzzy reasoning, Int. J. Mach. Learn. Cybern. (2021) 1–12.

[20] W. Xu, Q. Wang, X. Zhang, Multi-granulation fuzzy rough sets in a fuzzy tolerance approximation space, Int. J. Fuzzy Syst. 13 (4) (2011) 246–259.

[21] G. Lin, J. Liang, Y. Qian, An information fusion approach by combining multigranulation rough sets and evidence theory, Inf. Sci. 314 (2015) 184–199.

[22] J. Pearl, Evidential reasoning using stochastic simulation of causal models, Artif. Intell. 32 (2) (1987) 245–257.

[23] A. Tan, W.-Z. Wu, Y. Tao, A unified framework for characterizing rough sets with evidence theory in various approximation spaces, Inf. Sci. 454 (2018) 144–160.

[24] B. Sun, W. Ma, B. Li, X. Li, Three-way decisions approach to multiple attribute group decision making with linguistic information-based decision-theoretic rough fuzzy set, Int. J. Approx. Reason. 93 (2018) 424–442.

[25] Y. Yao, Probabilistic rough set approximations, Int. J. Approx. Reason. 49 (2) (2008) 255–271.

[26] Y. Yao, Three-way decisions with probabilistic rough sets, Inf. Sci. 180 (3) (2010) 341–353.

[27] Y. She, X. He, Y. Qian, W. Xu, J. Li, A quantitative approach to reasoning about incomplete knowledge, Inf. Sci. 451 (2018) 100–111.

[28] J. Tenenbaum, T. Griffiths, C. Kemp, Theory-based Bayesian models of inductive learning and reasoning, Trends Cogn. Sci. 10 (7) (2006) 309–318.

[29] Z. Yang, S. Bonsall, J. Wang, Fuzzy rule-based Bayesian reasoning approach for prioritization of failures in fmea, IEEE Trans. Reliab. 57 (3) (2008) 517–528.

[30] Q. Guo, Y. Qian, X. Liang, Mining logic patterns from visual data, in: International Conference on Data Mining Workshops, 2019, pp. 620–627.

[31] Q. Guo, Y. Qian, X. Liang, Y. She, D. Li, J. Liang, Logic could be learned from images, Int. J. Mach. Learn. Cybern. 12 (12) (2021) 3397–3414.

[32] Q. Guo, Y. Qian, X. Liang, J. Chen, H. Cheng, Multi-granulation multi-scale relation network for abstract reasoning, Int. J. Mach. Learn. Cybern. (2022), https://doi.org/10.1007/s13042-021-01484-5.

[33] B.M. Lake, R. Salakhutdinov, J.B. Tenenbaum, Human-level concept learning through probabilistic program induction, Science 350 (6266) (2015) 1332–1338.

[34] M. Segler, M. Waller, Neural symbolic machine learning for retrosynthesis and reaction prediction, Chemistry 23 (25) (2017) 5966–5971.

[35] I. Donadello, L. Serafini, A. Garcez, Logic tensor networks for semantic image interpretation, in: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17, 2017, pp. 1596–1602.

[36] S.N. Tran, A.S.d. Garcez, Deep logic networks: inserting and extracting knowledge from deep belief networks, IEEE Trans. Neural Netw. Learn. Syst. 29 (2) (2016) 246–258.

[37] Y. Hoshen, S. Peleg, Visual learning of arithmetic operation, in: Association for the Advancement of Artificial Intelligence, Phoenix, USA, 2016, pp. 3733–3739.

[38] Q. Hu, D. Yu, J. Liu, C. Wu, Neighborhood rough set based heterogeneous feature subset selection, Inf. Sci. 178 (18) (2008) 3577–3594.

[39] Y. Qian, H. Cheng, J. Wang, J. Liang, W. Pedrycz, C. Dang, Grouping granular structures in human granulation intelligence, Inf. Sci. 382 (2017) 150–169.

[40] C. Wang, Y. Huang, W. Ding, Z. Cao, Attribute reduction with fuzzy rough self-information measures, Inf. Sci. 549 (2021) 68–86.

[41] L. Zadeh, Some reflections on soft computing, granular computing and their roles in the conception, design and utilization of information/intelligent systems, Soft Comput. 2 (1) (1998) 23–25.

[42] T.Y. Lin, et al., Granular computing on binary relations ii: rough set representations and belief functions, in: Rough Sets in Knowledge Discovery, vol. 1, 1998, pp. 122–140.

[43] L. Zadeh, Fuzzy logic equals computing with words, IEEE Trans. Fuzzy Syst. 4 (2) (1996) 103–111.

[44] L.A. Zadeh, Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic, Fuzzy Sets Syst. 90 (90) (1997) 111–127.

[45] R.R. Yager, D.P. Filev, Fuzzy rule based models and approximate reasoning, in: Fuzzy Systems, Springer, 1998, pp. 91–133.

[46] J. Qian, C. Liu, D. Miao, X. Yue, Sequential three-way decisions via multi-granularity, Inf. Sci. 507 (2020) 606–629.

[47] Y. Qian, J. Liang, W. Pedrycz, C. Dang, Positive approximation: an accelerator for attribute reduction in rough set theory, Artif. Intell. 174 (9–10) (2010) 597–618.

[48] Y. Yao, Y. She, Rough set models in multigranulation spaces, Inf. Sci. 327 (2016) 40–56.

[49] P. Zhang, T. Li, Z. Yuan, L. Chuan, G. Wang, J. Liu, S. Du, A data-level fusion model for unsupervised attribute selection in multi-source homogeneous data, Inf. Fusion 80 (2022) 87–103.

[50] P. Zhang, T. Li, C. Luo, G. Wang, Amg-dtrs: adaptive multi-granulation decision-theoretic rough sets, Int. J. Approx. Reason. 140 (2022) 7–30.
[51] Y. Qian, J. Liang, Y. Yao, C. Dang, MGRS: a multi-granulation rough set, Inf. Sci. 180 (6) (2010) 949–970.
[52] Y. Qian, X. Liang, G. Lin, Q. Guo, J. Liang, Local multigranulation decision-theoretic rough sets, Int. J. Approx. Reason. 82 (2017) 119–137.
[53] H. Xiao, K. Rasul, R. Vollgraf, Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, CoRR, arXiv:1708.07747 [abs].
[54] Z.-H. Zhou, Abductive learning: towards bridging machine learning and logical reasoning, Sci. China Inf. Sci. 62 (7) (2019) 1–3.
[55] W. Dai, Q. Xu, Y. Yu, Z. Zhou, Bridging machine learning and logical reasoning by abductive learning, in: Proceedings of the 33rd International Conference on Neural Information Processing Systems, 2019, pp. 2815–2826.
[56] R. Manhaeve, S. Dumancic, A. Kimmig, T. Demeester, L. De Raedt, Deepproblog: neural probabilistic logic programming, in: Advances in Neural Information Processing Systems, vol. 31, 2018, pp. 3749–3759.
[57] A. Gaunt, M. Brockschmidt, N. Kushman, D. Tarlow, Differentiable programs with neural libraries, in: Proceedings of the 34th International Conference on Machine Learning, 2017, pp. 1213–1222.
[58] H. Li, L. Zhang, B. Huang, X. Zhou, Sequential three-way decision and granulation for cost-sensitive face recognition, Knowl.-Based Syst. 91 (2016) 241–251.
[59] Y. Qian, F. Li, J. Liang, B. Liu, C. Dang, Space structure and clustering of categorical data, IEEE Trans. Neural Netw. Learn. Syst. 27 (10) (2015) 2047–2059.
[60] Y. Yao, Three-way decision and granular computing, Int. J. Approx. Reason. 103 (2018) 107–123.
[61] W. Ding, J. Wang, J. Wang, Multigranulation consensus fuzzy-rough based attribute reduction, Knowl.-Based Syst. 198 (2020) 105945.
[62] B. Huang, C. xiang Guo, Y. liang Zhuang, H. xiong Li, X. zhong Zhou, Intuitionistic fuzzy multigranulation rough sets, Inf. Sci. 277 (2014) 299–320.
[63] G. Lin, J. Liang, Y. Qian, Multigranulation rough sets: from partition to covering, Inf. Sci. 241 (2013) 101–118.
[64] W. Xu, Y. Guo, Generalized multigranulation double-quantitative decision-theoretic rough set, Knowl.-Based Syst. 105 (2016) 190–205.
[65] X. Yang, X. Song, Z. Chen, J. Yang, On multigranulation rough sets in incomplete information system, Int. J. Mach. Learn. Cybern. 3 (3) (2012) 223–232.
[66] Á.E. Eiben, R. Hinterding, Z. Michalewicz, Parameter control in evolutionary algorithms, IEEE Trans. Evol. Comput. 3 (2) (1999) 124–141.
[67] A. Graves, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[68] G. Hinton, R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.
[69] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.