Guiding Mathematical Reasoning via Mastering Commonsense Formula Knowledge

Jiayu Liu
School of Data Science, University of Science and Technology of China & State Key Laboratory of Cognitive Intelligence
Hefei, China
jy251198@mail.ustc.edu.cn

Zhenya Huang*
School of Computer Science and Technology, University of Science and Technology of China & State Key Laboratory of Cognitive Intelligence
Hefei, China
huangzhy@ustc.edu.cn

Zhiyuan Ma
School of Computer Science and Technology, University of Science and Technology of China & State Key Laboratory of Cognitive Intelligence
Hefei, China
zhymma2000@gmail.com

Qi Liu
School of Computer Science and Technology, University of Science and Technology of China & State Key Laboratory of Cognitive Intelligence
Hefei, China
qiliuql@ustc.edu.cn

Enhong Chen*
Anhui Province Key Laboratory of Big Data Analysis and Application, University of Science and Technology of China & State Key Laboratory of Cognitive Intelligence
Hefei, China
cheneh@ustc.edu.cn

Tianhuang Su
OPPO Mobile Telecommunications
Shenzhen, China
sutianhuang@oppo.com

Haifeng Liu
University of Science and Technology of China
Hefei, China
bladehliu@gmail.com

ABSTRACT

Math formulas (e.g., "$distance = speed \times time$") serve as one of the fundamental commonsense knowledge in human cognition, where humans naturally acquire and manipulate them in logical thinking for mathematical reasoning problems. However, existing reasoning models mainly focus on learning heuristic linguistics or patterns to generate answers, but do not pay enough attention on learning with such formula knowledge. Thus, they are not transparent (thus uninterpretable) in terms of understanding and grasping basic mathematical logic. In this paper, to promote a step forward in the domain, we first construct two datasets (Math23K-F and MAWPS-F) with precise annotations of formula usage in each reasoning step for math word problems. Especially, our datasets are refined on the benchmark datasets, and thus ensure the generality and comparability for relevant research. Then, we propose a novel Formula-mastered Solver (FOMAS) with the guidance of mastering formula knowledge to solve the problems. Specifically, we establish FOMAS with two systems drawing insight from the dual process theory, including a Knowledge System and a Reasoning System, to learn and apply formula knowledge, respectively. The Knowledge System accumulates the math formulas, where we propose a novel pretraining manner to mimic how humans grasp the mathematical logic behind them. Then, in the Reasoning System, we develop elaborate formula-guided symbol prediction and goal generation methods that retrieve the necessary formula knowledge from Knowledge System to improve both reasoning accuracy and interpretability. It organically simulates how humans conduct complex reasoning under the explicit instruction of math formulas. Experimental results prove that FOMAS has a stronger reasoning ability and achieves a more interpretable reasoning process, which verifies the necessity of introducing formula knowledge transparently.

CCS CONCEPTS

• **Computing methodologies → Knowledge representation and reasoning**; **Symbolic and algebraic manipulation**.

KEYWORDS

Knowledge Representation and Reasoning, Math Word Problem

ACM Reference Format:
Jiayu Liu, Zhenya Huang, Zhiyuan Ma, Qi Liu, Enhong Chen, Tianhuang Su, and Haifeng Liu. 2023. Guiding Mathematical Reasoning via Mastering Commonsense Formula Knowledge. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA*. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3580305.3599375

[*] Corresponding Authors

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

KDD '23, August 6–10, 2023, Long Beach, CA, USA.

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0103-0/23/08…$15.00
https://doi.org/10.1145/3580305.3599375

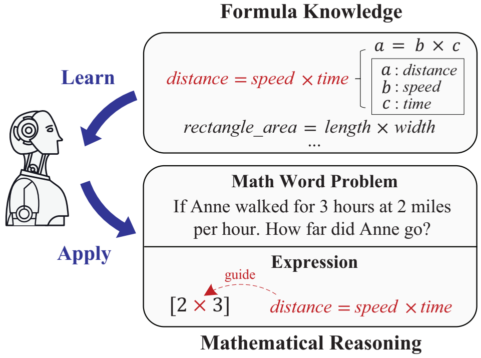

Formula Knowledge
Learn
$distance = speed \times time$
$a = b \times c$
a : distance
b : speed
c : time
$rectangle\_area = length \times width$
...
Apply
Math Word Problem
If Anne walked for 3 hours at 2 miles per hour. How far did Anne go?
Expression
guide
[2 × 3]
$distance = speed \times time$
Mathematical Reasoning

Figure 1: Example of learning and applying the formula knowledge to mathematical reasoning on MWP.

1 INTRODUCTION

Knowledge is the human belief encompassing empirical awareness of facts and epistemic contact with the world [45]. It lays the foundation for human cognition, where humans naturally acquire knowledge from experience and manipulate it in cognitive behaviors [33, 44]. In the domain of data mining, there is an emerging trend to investigate how to gain knowledge from data and apply it in several complex reasoning tasks [15, 31, 32, 39]. Among them, mathematical reasoning is one of the core tasks in quest of models with human cognitive ability, which requires various knowledge (e.g., algebra, geometry, probability) in logical thinking [5, 16, 53].

Specially, math formula is a kind of essential and commonsense knowledge formed from human experience [25]. We are constantly learning and applying various math formulas to figure out problems, and typically, math word problems (MWP). Let us take an example in Figure 1. Solving the MWP generally asks to read a problem description ("If Anne ... go ?") and then reasons an expression ("2 × 3") for getting the answer. This process requires necessary formulas ("$distance = speed \times time$") to explicitly serve as the guidance that navigates the thinking pattern and directs symbol derivation based on problem understanding [12, 14], which reflects the complicated mastery of abstract logic in human cognition. Thus, the mastery of math formulas for conducting reasoning like humans is a necessary sign of the intelligent level of models [6].

However, in the literature, such explicit knowledge has not attracted enough attention in relevant research on reasoning. Taking the widely studied MWP as an example, although traditional work takes into account mathematical rules or templates [2, 11, 37], they operate according to manually defined programs without really gaining the ability to understand this knowledge. Comparatively, most current methods rely on a Seq2Seq model [35, 47], which generate a hidden state (or named as reasoning goal [49]) at each decoding step to predict a specific symbol. They mainly grasp the linguistic semantics [22, 27] and heuristic reasoning patterns [19, 49], but ignore whether the model can master the epitome of logic manifested in explicitly utilizing formula knowledge [38, 50].

To further verify the necessity, we tend to experimentally quantify the mastery degree of formula knowledge for current models. Nevertheless, existing mathematical benchmark datasets (e.g., Math23K [47], MAWPS [24], GSM8K [4], MATH [13]) do not provide such a label for formula usage in expression reasoning process. In this paper, to promote a step forward in the domain, we first construct two datasets (Math23K-F and MAWPS-F) with precise annotations to support specific explorations. They contain 51 and 18 different math formulas and 23, 162 and 2, 373 annotated problems refined on benchmarks Math23K and MAWPS respectively, which ensures the generality and comparability for relevant research. For each problem, we annotate the formula required at each reasoning step in its expression, e.g., "$distance = speed \times time$" for "×" in Figure 1. Based on our datasets, we conduct empirical experiments (described in Section 3.1) for existing representative methods (GTS [49], Graph2Tree [54], BERT-Tree [26]) and observe that more than 22% of their errors are due to the inability to use formulas. Thus, we argue that it is vital to investigate how models learn and apply formula knowledge for mathematical reasoning.

Along this line, there remain many technical challenges to be confronted. First, math formulas are highly symbolic that involve abstract structure and concrete concepts. For example, "$distance = speed \times time$" in Figure 1 is composed of the structure "$a = b \times c$" and concepts "$distance, speed, time$". "$a = b \times c$" reflects the calculation pattern that could be shared between different formulas (e.g., "$rectangle\_area = length \times width$" also satisfies "$a = b \times c$"), while concepts determine the semantic information that should be displayed in the structure to truly understand the formula. It is indispensable to mine their respective meanings as well as integrate them as a whole. Second, math formulas contain rich mathematical logic. Humans not only learn whether a formula is correct or not, but are also able to perform logical transformations such as changing "$distance = speed \times time$" to "$speed = distance \div time$". Teaching models to master such implied knowledge is necessary but challenging. Third, the process of human application of formula knowledge is sophisticated, which includes selecting formulas according to different problems (e.g., choose "$distance = speed \times time$" when reasoning "×" for problem in Figure 1) and utilizing them in multiple reasoning steps (inherit the information of "$speed$" and "$time$" to derive symbols "2" and "3"). The unclear cognitive mechanisms behind it brings many difficulties in modeling.

To address the challenges above, we propose a novel Formula-mastered Solver (FOMAS) to mimic how humans solve MWP under the guidance of formula knowledge. In FOMAS, we draw insights from the dual process theory [8, 20] that elaborates on the structure of human cognition to construct two systems: Knowledge System and Reasoning System. The Knowledge System plays the role of human brain to store, represent, and learn formula knowledge, while the Reasoning System analytically applies the formula knowledge accumulated in Knowledge System to reason answers for MWP. Specifically, in Knowledge System, we first adequately excavate the information of each formula by decoupling and encoding its structure feature and lexical feature. Then, we propose a pretraining manner to realize autonomous formula learning, where two novel pretraining objectives are designed based on human understanding of formula legality and flexibility. In Reasoning System, given a math word problem, we retrieve formulas from the Knowledge System to guide the expression decoding phase (i.e., symbol prediction and goal generation). To be specific, we develop three sophisticated reasoning mechanisms to predict a symbol at each reasoning step, including a formula-selected one, a formula-inherited one, and a typical direct one. Their results are organically ensembled to simulate complex human thought. Moreover, we introduce the representations of intermediate steps of formulas into deriving explainable reasoning goals. Extensive experiments on our benchmark datastes verify FOMAS' improvements on answer accuracy and reasoning interpretability. The contributions of this paper are:

- We propose a Formula-mastered Solver (FOMAS) that learns and applies formula knowledge to conduct mathematical reasoning. To the best of our knowledge, we are the first to explore it by referring to the characteristics of human cognition, whose main ideas of Knowledge-Reasoning systems and specific learning/reasoning methods can be quite general for different types of mathematical problems.
- We design a novel pretraining manner to learn the knowledge behind math formulas and develop elaborate formula-guided symbol prediction/goal generation mechanisms, which enhance reasoning accuracy and interpretability.
- We construct two benchmark MWP datasets (Math23K-F, MAWPS-F) with annotations of the required formula at each reasoning step to support the exploration of formula knowledge in the domain. Extensive experiments on them clearly demonstrate the effectiveness of FOMAS.

2 RELATED WORK

Math Word Problem. Mathematical reasoning aims to acquire knowledge from data and apply it to solve math problems. According to different problem types, representative benchmarks include for MWP (e.g., Math23K [47]), geometry (e.g., Inter-GPS [34]), unified task (e.g., MATH [13]), etc., among which MWP motivates the fundamental and far-reaching research since the 1960s [10, 17].

In the literature, traditional MWP solvers can be classified into three types: rule-based [2], statistic-based [37], and semantics parsing-based [43]. They require manually created schemas, templates, or formal language to derive expressions, respectively, and thus suffer from limited applications on large datasets and low generality. Recently, Wang et al. [47] presented a Seq2Seq model that adopted the encoder-decoder to directly translate a problem into the expression. Based on such a structure, most advanced work can be summarized from two aspects: improving the semantic understanding ability of encoder [22, 28, 29] and the reasoning ability of decoder [19, 49]. For problem understanding, Zhang et al. [54] proposed a GCN-based encoder to capture the relationships and order information of quantities, Lin et al. [29] adopted a word-clause-problem hierarchical encoder to simulate human reading habits. Besides, pretrained language models [22, 26, 27] and external knowledge graphs [48] have also been applied to inject human knowledge into problem comprehension. As for reasoning, Xie et al. [49] adopted a goal-driven decomposition mechanism to generate binary expression trees, while Wang et al. [46] designed M-tree structure to unify the diverse outputs, and Jie et al. [19] proposed a deductive reasoner to iteratively construct step-by-step expressions.

Neural-Symbolic Systems. The core goal of our paper is to simulate human cognition to master the symbolic math formula knowledge for mathematical reasoning. Thus, we report the related work regarding neural-symbolic systems [51], which have flourished in many reasoning tasks such as knowledge graph reasoning [3, 42], visual reasoning [36, 52], and text reasoning [30, 39]. For example, Qu et al. [42] proposed pLogicNet that leveraged the first-order logic of triplets to support inference on knowledge graphs. Mao et al. [36] constructed NS-CL that executed logic operations in a symbolic program to derived answers for visual questions. Specially, for mathematical reasoning, Peng et al. [39] designed a novel representation architecture GATE to conduct interpretable symbolic deduction and computation. Aiming at MWP, Qin et al. [41] proposed NS-Solver to incorporate symbolic constraints in training by four auxiliary tasks. Yang et al. [50] built LogicSolver that used logic formula as prompts to enrich problem understanding and predicted formulas as explanations, which for the first time introduced and verified the importance of formula knowledge for MWP.

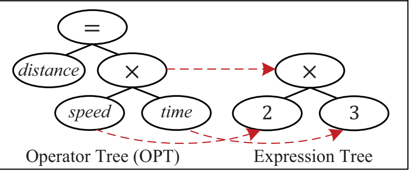
Problem	If Anne walked for 3 hours at 2 miles per hour. How far did Anne go?
Answer	6
Expression	[2 × 3]
Formula	[$\emptyset$, *distance* = *speed* × *time*, $\emptyset$]
Explanation	 = distance × × speed time 2 3 Operator Tree (OPT) Expression Tree

Figure 2: An example of our annotated data.

Our work improves previous studies from the following three aspects. First, current neural-symbolic systems mainly focus on logic rules and knowledge graph. Comparatively, we explore how to master the commonsense but essential formula knowledge in mathematical reasoning, aiming at which we propose a novel formula pretraining schema and design elaborate application mechanisms. Second, compared with existing work on MWP, our method learns and applies the math formulas to finely guide expression reasoning, which is more transparent and accurate in terms of the reasoning process. Specially, unlike the most relevant LogicSolver [50] that uses formulas as prompts and post-explanation, our formula application mechanisms are more in line with human cognition and achieves better effectiveness. Third, from the perspective of dataset, our benchmarks provide high-quality annotations to support exploration of formula knowledge, maintaining great generality and comparability to potentially boost further study in the domain.

3 PRELIMINARIES

3.1 Dataset Construction and Analysis

In the field of math word problem (MWP), representative datasets include Math23K [47], MAWPS [24], SVAMP [38], GSM8K [4], etc.. However, to the best of our knowledge, they do not provide satisfactory annotations to guide models how to apply formula knowledge step by step in the reasoning process. In response to such a burgeoning need, we expect to construct datasets that have two characteristics. First, they should precisely describe the formula usage of reasoning process, i.e., annotate the formula applied at each reasoning step. Second, we hope to ensure the generality so that most previous models can be easily and fairly compared on them. Based on these considerations, we decide to annotate the two most widely studied MWP datasets Math23K [47] and MAWPS [24],

Table 1: The 5 most frequently used math formulas.

Math23K-F	1. $distance = speed \times time$ 2. $work = rate \times time$ 3. $total\_cost = unit\_cost \times total\_number$ 4. $total\_amount = unit\_amount \times total\_number$ 5. $total\_weight = unit\_weight \times total\_number$
MAWPS-F	1. $total\_amount = unit\_amount \times total\_number$ 2. $total\_cost = unit\_cost \times total\_number$ 3. $total\_income = unit\_income \times total\_number$ 4. $distance = speed \times time$ 5. $work = rate \times time$

which contain 23,162 and 2,373 problems at the elementary school level respectively. There original data consists of "problem", "answer", and "expression", as shown in the top three lines in Figure 2.

Our annotation is conducted as follows. Inspired by existing work [1, 50], we firstly collected essential math formulas from textbooks and summarized 51 and 18 representative ones on Math23K and MAWPS respectively under the guidance of two elementary school teachers. Secondly, we invited five well-trained annotators with undergraduate degree to manually select the most suitable formula for each reasoning step of data in Math23K and MAWPS from the set of all formulas and their variants (will be defined in Section 3.2). For example, as depicted in the "Formula" line in Figure 2, the annotation of problem "If Anne ... go ?" is [ϕ, $distance = speed \times time$, ϕ], whose length is equal to the length of expression "2×3". The explanation behind it is that reasoning the "×" requires "$distance = speed \times time$", while reasoning "2" and "3" do not need additional formulas (thus denoted as an empty formula ϕ). Thirdly, another three annotators were asked to evaluate the annotations, based on which the annotators modified the results and repeated the evaluation-modification processes. After three revisions with a pass rate of 93.2%, 96.3%, and 97.7%, we settled on the final annotations and name the new datasets as Math23K-F and MAWPS-F. In summary, on our benchmarks, 33.5% and 38.4% of problems require the use of formulas respectively. We report the 5 most frequently used formulas in Table 1 and summarize the dataset statistics in Table 2. More analyses are presented in *Appendix A*.

Based on our benchmark datasets, we tend to verify whether the study of formula knowledge is necessary. For this purpose, we select the three most representative SOTA MWP models, including GTS [49], Graph2Tree [54], and BERT-Tree [26], and evaluate their mastery degree of formula knowledge. Specifically, we introduce a *PIF* metric that calculates the **P**ercentage of problems that a model answers **I**ncorrectly at steps requiring a **F**ormula (as we annotated above) in all problems that it gets the answer wrong. A higher *PIF* reflects a greater defect in formula mastery. As shown in Figur 3, *PIF* of the three models exceeds 0.220 on both datasets, which indicates that the accuracy of the existing work is greatly limited due to lack of applying math formulas. Thus, we argue that it is necessary to explore how to master formula knowledge for conducting more reliable mathematical reasoning.

3.2 Problem Definition

Formally, the input of a math word problem P is a sequence of n words and numeric values: $X_P = [x_1, x_2, ..., x_n]$, where x_i is either a word (e.g., "If", "Anne" in Figure 2) or a number (e.g., "3", "2"). The output of P is a sequence of m symbols: $Y_P = [y_1, y_2, ..., y_m]$. Symbol y_i is taken from a vocabulary V_P composed of the operator set V_O (e.g., $\{+, \times, -, \div\}$), numeric constant set V_N (e.g., $1, \pi$), and numeric values N_P in X_P, i.e., $V_P = V_O \cup V_N \cup N_P$. As N_P varies with the input sequence, different problems may have different V_P. The target of MWP is to train a model that reads the problem description X_P and then generates the corresponding expression Y_P, based on which calculates a numeric answer for P (e.g., "6").

Table 2: Statistics of our benchmark datasets.

Dataset	Math23K-F	MAWPS-F
Num. problems	23,162	2,373
Num. formulas (and variants)	51 (131)	18 (46)
Num. problems requiring formula	7,750	911
Avg. problem length	28.02	30.08
Avg. expr. length	5.55	4.20

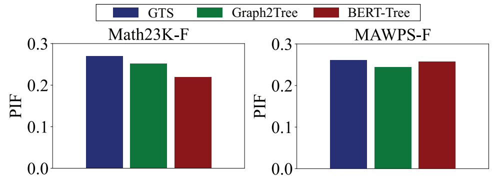


Figure 3: *PIF* **of GTS, Graph2Tree, and BERT-Tree.**

For formula knowledge, we formalize it as a set of K math formulas $R = \{r_1, r_2, ..., r_K\}$ ($K = 51$ and 18 on Math23K-F and MAWPS-F respectively). Each formula r_k is represented as an Operator Tree (OPT) [40]. For example, the formula "$distance = speed \times time$" corresponds to a tree that contains five nodes as shown in Figure 2, with "=" as the root and "$distance$""$speed$""$time$" as the leaf nodes. Thus, we denote r_k as a sequence of l elements $r_k = [z_1, z_2, ..., z_l]$ that represents the prefix expression of the OPT, where z_i is either an abstract concept (e.g., "$time$") or an operator (e.g., "×"). For example, "$distance = speed \times time$" is represented as [=, $distance$, ×, $speed$, $time$]. Furthermore, we define a set of variants $A(r) \not\subseteq R$ for each formula r (e.g., "$speed = distance \div time$" and "$time = distance \div speed$") to imply logical formula transformation, which is important in formula learning and application in Section 4.1 and Section 4.2. In summary, the research problem in this paper is:

Definition 3.1. Given the MWP dataset $D = \{(X_P, Y_P)\}$ and the formula set R, our goal is to build a model that learns the formula knowledge in R and applies it to reason the expression $Y_P = [y_1, y_2, ..., y_m]$ for each problem $X_P = [x_1, x_2, ..., x_n]$.

4 FORMULA-MASTER SOLVER

Generally, the dual process theory [8, 20] indicates that human cognition is built on two systems: one is an intuitive, unconscious system that processes human experience and knowledge, and the other is a reasoning system that uses the knowledge for slow, explicit logical thinking. Drawing this insight, we construct two systems in our Formula-master Solver (FOMAS): Knowledge System and Reasoning System, as shown in Figure 4(a)(b). Specifically, Knowledge System stores the formula knowledge R and mimics how humans represent and learn it autonomously, while Reasoning System applies the formula knowledge in Knowledge System to reason the expression Y_P for a specific math word problem X_P.

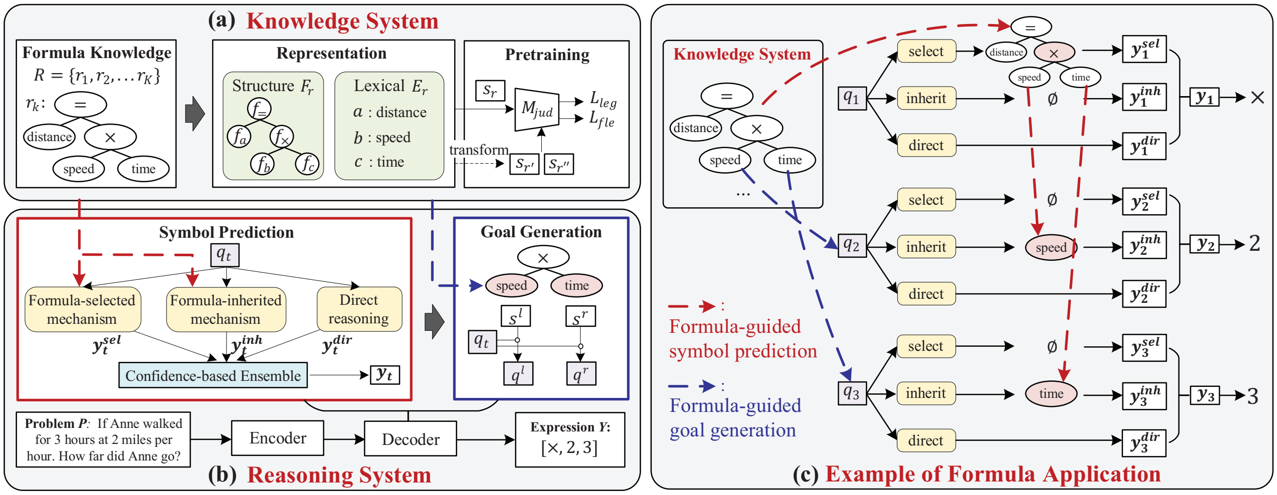


Figure 4: The overview of FOMAS. The left part (a)(b) shows the architecture of FOMAS that consists of Knowledge System and Reasoning System . The right part (c) is an example workflow of formula-guided symbol prediction and goal generation.

4.1 Knowledge System

In this system, we first design a representation module to comprehensively mine the features of formulas. Then we propose a pretraining manner to learn these features, which resembles humans' mastery of the mathematical logic behind formulas.

4.1.1 Formula Representation. As logicians and mathematicians have pointed out, the difficulty of understanding formula knowledge lies in distinguishing formulas that have the same structure but different meanings [7]. For example, "*distance* = *speed* × *time*" and "*rectangle_area* = *length* × *width*" have the same structure abstracted as $a=b\times c$. However, they are different in the lexicons of a,b,c (e.g., $a=distance$, $b=speed$, $c=time$). The meaning of the whole formula is compositionally built up from the combination of these two types of information, both of which are necessary. Inspired by it, we decouple each formula $r\in R$ into two types of features: structural feature F_r and lexical feature E_r, which focus on the structural information and concrete meaning respectively.

Specifically, the structural feature F_r defines the order and mode of formula calculation, so we design an innovative kind of node function f and formalize F_r as a sequence of it, i.e., $F_r\triangleq[f_1,f_2,\dots,f_w]$, where w is the length of formula r and f_i is the function of i-th node (i.e., z_i). For example, the structural feature of "*distance* = *speed* × *time*" is $F_r=[f_=,f_a,f_\times,f_b,f_c]$. The input and output of f will be explained below after introducing the lexical feature.

The lexical feature E_r reflects the semantics of formula r embodied in the leaf nodes of its OPT (e.g., a,b,c). We formalize it as a sequence of concept vectors, i.e., $E_r\triangleq[\boldsymbol{e}_1,\boldsymbol{e}_2,\dots,\boldsymbol{e}_v]$, where v is the number of leaf nodes, $\boldsymbol{e}_i\in\mathbb{R}^d$ is the i-th node vector, and d is the dimension. For example, the lexical feature of "*distance* = *speed* × *time*" is $E_r=[\boldsymbol{e}_1,\boldsymbol{e}_2,\boldsymbol{e}_3]$, where $\boldsymbol{e}_1,\boldsymbol{e}_2,\boldsymbol{e}_3$ are the concept vectors of "*distance*", "*speed*", "*time*", respectively.

Now we explain the computing flow of the node functions, which basically follows a bottom-up pattern. Specifically, given formula r, for a leaf node in its OPT, the node function f is an identity mapping, whose input and output are the same concept vector of the node. For example, as marked green in Figure 4(a), $f_a(\cdot)\triangleq f_a(\boldsymbol{e}_a)=\boldsymbol{e}_a$, where $\boldsymbol{e}_a$ is input as the concept vector of "*distance*" (i.e., $\boldsymbol{e}_1$) for "*distance* = *speed* × *time*". For a non-leaf node, the function f encodes a representation of the intermediate step in the formula. It takes into the outputs of its left and right child nodes, and fuses them with the operator information. For example, in Figure 4(a), $f_\times(\cdot,\cdot)\triangleq f_\times(f_b(\boldsymbol{e}_b),f_c(\boldsymbol{e}_c))$. By taking $\boldsymbol{e}_b=\boldsymbol{e}_2,\boldsymbol{e}_c=\boldsymbol{e}_3$, we can obtain the embedding of node "×", which reflects the meaning of "*speed* × *time*". Specially, $f_\cdot(\cdot,\cdot)$ is implemented as a network:

$$\begin{aligned}f_\tau(\boldsymbol{e}_l,\boldsymbol{e}_r)&=\boldsymbol{v}_\tau\odot\boldsymbol{g}_\tau,\\ \boldsymbol{v}_\tau=\sigma(\boldsymbol{W}_{n1}\cdot[\boldsymbol{e}_l,\boldsymbol{e}_r,\boldsymbol{o}_\tau]),\ &\boldsymbol{g}_\tau=tanh(\boldsymbol{W}_{n2}\cdot[\boldsymbol{e}_l,\boldsymbol{e}_r,\boldsymbol{o}_\tau]),\end{aligned}\tag{1}$$

where $\tau\in\{+,-,\times,\div,=\}$ indicates the operator, $\boldsymbol{o}_\tau\in\mathbb{R}^d$ is its symbol embedding, $\boldsymbol{e}_l,\boldsymbol{e}_r$ are the outputs of the left and right child nodes (e.g., $f_b(\boldsymbol{e}_b),f_c(\boldsymbol{e}_c)$), $\odot$ is element-wise product, σ is the sigmoid function, and $\boldsymbol{W}_{n1},\boldsymbol{W}_{n2}$ are parameters.

In summary, structural feature $F_r=[f_1,f_2,\dots,f_w]$ defines the calculation pattern of formula r by node functions, while lexical feature $E_r=[\boldsymbol{e}_1,\boldsymbol{e}_2,\dots,\boldsymbol{e}_v]$ stores the practical meanings by concept vectors. Combining them, we can iteratively encode a semantic embedding s for each non-leaf node by Eq. (1), which represents an intermediate state of the formula. We denote these semantic embeddings as $S_r=[\boldsymbol{s}_1,\dots,\boldsymbol{s}_{w-v}]$, where $\boldsymbol{s}_1$ is the embedding of root "=" that encodes the information of the whole formula. In Section 4.2, we will see how they contribute to guiding mathematical reasoning, which reflects the superiority of our representation module over directly encoding formulas by RNN-like models.

4.1.2 Formula Pretraining. In order to mimic how humans process and learn the formula knowledge autonomously, we further pretrain FOMAS' representation module (i.e., parameters of node functions in F_r and concept vectors in E_r). However, since the formulas are short in length, strong in logic, and do not have a large corpus, it is not suitable to adopt the existing pretraining manners (e.g., Masked Language Model [56]). Thus, as shown in Figure 4(a), to embody mathematical logic into formula learning, we design two novel pretraining objectives: legality L_{leg} and flexibility L_{fle}.

The legality L_{leg} focuses on learning to distinguish whether a formula is legal or not. That is, not only do humans acquire that "*distance* = *speed* × *time*" is a valid formula, but also know "*distance* = *speed*×*width*" is invalid and avoid using it in reasoning. Through this objective, FOMAS can pay attention to the detailed components of formulas and grasp more refined understanding. For this purpose, we introduce a dichotomous task. For each formula $r \in R$, we first obtain the semantic vector of its root node "=" using features F_r and E_r according to Section 4.1.1, denoted as $\boldsymbol{s}_r$. Then, we randomly replace one of the concept vectors in E_r to construct an illegal formula sample r', e.g., "*distance* = *speed* × *width*", and obtain the representation of its root $\boldsymbol{s}_{r'}$ similarly. Finally, we input $\boldsymbol{s}_r, \boldsymbol{s}_{r'}$ to calculate the following loss:

$$L_{leg} = \sum_{r \in R} l_{BCE}(M_{jud}(\boldsymbol{s}_r), 1) + l_{BCE}(M_{jud}(\boldsymbol{s}_{r'}), 0), \\ M_{jud}(\boldsymbol{s}) = \sigma(\boldsymbol{W}_{u1} \cdot ReLu(\boldsymbol{W}_{u2} \cdot tanh(\boldsymbol{W}_{u3} \cdot \boldsymbol{s}))), \tag{2}$$

where $M_{jud}(\boldsymbol{s}) : \mathbb{R}^d \rightarrow [0, 1]$ is a judgment network to learn the probability that r and r' are legal. l_{BCE} is the BCE loss, 1 and 0 are ground-truth labels that represent "legal" and "illegal", respectively.

For a formula, it is not enough to grasp its own legitimacy, because humans can based on it to judge whether its complex transformations (e.g., "*speed* = *distance* ÷ *time*" and "*time* = *distance* ÷ *speed*" for "*distance* = *speed* × *time*") are also legal, which together constitute a comprehensive mastery of the mathematical logic. For this purpose, we further incorporate a flexibility objective L_{fle}, whose basic idea is that the variants of an (il)legal formula are also (il)legal. In specific, for formula $r \in R$, we first obtain its variants $A(r)$. Then, for each $r'' \in A(r)$, we similarly construct a negative sample r''_{-}. Finally, we feed them into M_{jud} and calculate:

$$L_{fle} = \sum_{r \in R} \sum_{r'' \in A(r)} l_{BCE}(M_{jud}(\boldsymbol{s}_{r''}), 1) + l_{BCE}(M_{jud}(\boldsymbol{s}_{r''_{-}}), 0). \tag{3}$$

With L_{leg} and L_{fle}, our pretraining schema is to minimize Eq. (4), which can capture strong mathematical logic in formula representations to better guide the reasoning process in Section 4.2.

$$L_{pretrain} = L_{leg} + L_{fle}. \tag{4}$$

4.2 Reasoning System

The Reasoning System aims at applying the formula knowledge in the Knowledge System to reason answers for a specific problem. As depicted in Figure 4(b), we generally adopt an encoder-decoder manner which first encodes the problem sentence and then decodes an expression tree [49]. This system mainly focuses on the decoding phase, which uses formulas to guide the symbol prediction and goal generation processes in decoder, namely formula-guided symbol prediction and formula-guided goal generation, respectively.

4.2.1 Encoder-Decoder. Given problem P, we first input the problem sentence $X_P = [x_1, x_2, ..., x_n]$ into an encoder f_θ to obtain the word representations $\boldsymbol{H} = [\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_n]$ and generate the initial reasoning goal $\boldsymbol{q}_1$ for the decoder:

$$(\boldsymbol{H}, \boldsymbol{q}_1) = f_\theta([x_1, x_2, ..., x_n]). \tag{5}$$

f_θ can be specified as RNN (e.g., GTS [49]), BERT (e.g., BERT-Tree [26]), or specific MWP encoder (e.g., HMS [29]). Here we adopt BERT [21] due to its strong capability of language modeling.

Then, we utilize a decoder to generate the expression $Y_P = [y_1, y_2, ..., y_m]$ step by step. At each step t ($t = 1, 2, ..., m$), the decoder 1) predicts the symbol y_t (e.g., "×", "2") given the reasoning goal $\boldsymbol{q}_t$ by $P(y_t|y_1, ..., y_{t-1}, \boldsymbol{q}_t, \boldsymbol{H})$; 2) generates the next reasoning goal $\boldsymbol{q}_{t+1}$ based on q_t and $e(y_t)$ to support the next step $t+1$, where $e(y_t)$ is the embedding of symbol y_t, which is taken as $\boldsymbol{h}_i$ if y_t is the word i in P or $\boldsymbol{o}_\tau$ if y_t is operator τ.

Intuitively, the formula knowledge can guide both symbol prediction and goal generation in decoder. For example, when solving the problem in Figure 4(c), if we figure out that the first step (i.e., $t = 1$) requires the use of formula "*distance* = *speed* × *time*", then we can easily extract the symbol "×" from it as y_1. Then, having decided to use this formula, we can clearly know that the $t = 2$ and $t = 3$ steps are to get the "*speed*" and "*time*" information respectively, which injects great interpretability into the reasoning goals $\boldsymbol{q}_2, \boldsymbol{q}_3$, so as to benefit further reasoning of $y_2 = 2, y_3 = 3$.

In summary, given problem sentence X_P and expression Y_P, the training objective of FOMAS is:

$$L = \sum_{P} \sum_{t} -log\ P(y_t|y_1, ..., y_{t-1}, \boldsymbol{q}_t, \boldsymbol{H}) + \alpha \cdot L_r \tag{6}$$

where $\sum_P \sum_t -logP(y_t|y_1, ..., y_{t-1}, \boldsymbol{q}_t, \boldsymbol{H})$ is the symbol prediction loss. L_r is a loss that corresponds to deciding the used formula as illustrated above, which will be specified in Section 4.2.2. α is a hyper-parameter that balances these two losses.

4.2.2 Formula-guided Symbol Prediction. Now we discuss how FOMAS applies the formula knowledge in the Knowledge System to predict symbol y_t given the reasoning goal $\boldsymbol{q}_t$. Specifically, we propose three types of symbolic reasoning mechanisms summarized from sophisticated human thought process [9], i.e., formula-selected mechanism, formula-inherited mechanism, and direct reasoning. Symbol prediction at each step t requires a combination of these three mechanisms, as depicted in Figure 4(b) and (c).

Formula-selected mechanism. Humans can naturally select a formula by referring to the current reasoning goal $\boldsymbol{q}_t$ at step t and extract a symbol from it as y_t. For example in Figure 4(c), if the goal $\boldsymbol{q}_1$ implies "calculate the distance", we can easily pick out the formula "*distance* = *speed* × *time*" and then reason "y_1 = ×", retrieving the semantic and symbolic information of formulas from Knowledge System respectively. Formally, at the current reasoning step t, we first calculate the attention of formula r and the goal $\boldsymbol{q}_t$ on problem word representations $\{\boldsymbol{h}_i\}$ by:

$$att_i^r = softmax(\boldsymbol{w}_a^\top \cdot tanh(\boldsymbol{W}_a \cdot [\boldsymbol{s}_r, \boldsymbol{h}_i])), \\ att_i^g = softmax(\boldsymbol{w}_b^\top \cdot tanh(\boldsymbol{W}_b \cdot [\boldsymbol{q}_t, \boldsymbol{h}_i])), \tag{7}$$

where $\boldsymbol{s}_r$ is the semantic embedding of the root "=" of r, and $\boldsymbol{w}_., \boldsymbol{W}_.$ are parameters. Then, we integrate these two kinds of attention to generate a score for r:

$$score(r) = \boldsymbol{W}_{s1} \cdot tanh(\boldsymbol{W}_{s2} \cdot \boldsymbol{c}_r),\ \boldsymbol{c}_r = (att_i^r + \beta \cdot att_i^g) \cdot \boldsymbol{H}. \tag{8}$$

Here, the hyper-parameter β controls the integration weight. Since a problem may not require any formula (e.g., calculate the sum of two values), we also consider the empty formula ϕ introduced in Section 3.1, denoted as r_0, and calculate $score(r_0)$ by Eqs. (7)(8), where $\boldsymbol{s}_{r_0}$ is a learnable vector. Finally, we derive the probability of selecting formula r from all formulas and their variants, denoted as

$\bar{R} = R \cup \{A(r), r \in R\} \cup \{r_0\}$ ($|\bar{R}| = 132$ and 47 on Math23K-F and MAWPS-F respectively according to the statistics in Table 2):

$$P_{sel}(r) = softmax(score(r)), r \in \bar{R}, \tag{9}$$

and extract the right child y of root "=" for the formula with the highest $P_{sel}(r)$ (e.g., "×" in Figure 4(c)). We take the one-hot embedding of y as the reasoning result, denoted as $y_t^{sel} \in \mathbb{R}^{|V_P|}$:

$$y_t^{sel} = one - hot(right\_child(argmax_{r \in \bar{R}}\, P_{sel}(r))). \tag{10}$$

To train all the parameters in Eqs. (7)(8), we define L_r in Eq. (6) as the following cross entropy, where $y_r \in \{0, 1\}$ is the ground-truth label of selected formula that we have annotated in Section 3.1:

$$L_r = -\sum_{r \in \bar{R}} y_r \cdot log\, P_{sel}(r). \tag{11}$$

Formula-inherited mechanism. The selected formula not only affects the current step, but also implies a kind of thinking pattern that navigates multiple reasoning steps [25]. For example, in Figure 4(c), after selecting the formula "$distance = speed \times time$" at $t = 1$, we can inherit the "$speed$" concept (i.e., the child of node "×" in formula OPT) at $t = 2$ to derive $y_2 = 2$ (find the "$speed$" value). Inspired by it, in this mechanism, we inherit the symbols and concepts of the formula selected in preceding reasoning steps to derive y_t. Formally, at step t: (1) if an operator (e.g., "×") is inherited, we obtain its one-hot embedding as $y_t^{inh} \in \mathbb{R}^{|V_P|}$, otherwise (2) if an abstract concept (e.g., "$speed$") is inherited, we reason y_t^{inh} based on its concept vector $\boldsymbol{e}_{inh}$ retrieved from Knowledge System:

$$y_t^{inh} = softmax(\boldsymbol{W}_{h1} \cdot tanh(\boldsymbol{W}_{h2} \cdot [\boldsymbol{e}_{inh}, \boldsymbol{q}_t])). \tag{12}$$

Direct Reasoning. Besides, humans can directly reason a symbol based on the reasoning goal $\boldsymbol{q}_t$ without any formula guidance as most existing MWP solvers do. Here, we set it as the scheme of GTS [49] and its reasoning result is denoted as $y_t^{dir} \in \mathbb{R}^{|V_P|}$.

To aggregate the results of different reasoning mechanisms, i.e., y_t^{sel}, y_t^{inh}, and y_t^{dir}, we further propose a confidence-based ensemble inspired by the mixture of experts (MoE) [18, 55]. Specifically, we hold that P_{sel} reflects the confidence of using the selected formula. Similarly, the inherited formula also comes with a probability that is inherited from the step selecting it, which we denote as P_{inh}. In summary, FOMAS reasons y_t by ensembling all mechanisms:

$$y_t = \frac{P_{sel} \cdot y_t^{sel} + P_{inh} \cdot y_t^{inh} + 1 \cdot y_t^{dir}}{P_{sel} + P_{inh} + 1}. \tag{13}$$

4.2.3 Formula-guided Goal Generation. After predicting y_t, the core step of FOMAS is to apply the formula knowledge to generate an explainable next reasoning goal $\boldsymbol{q}_{t+1}$. For this purpose, as shown in Figure 2, we observe that for the formula OPT and the expression tree, there is a one-to-one correspondence between the children of the same node. For instance, for "×", its left child "$speed$" in formula conveys the goal to reason the left child "2" in the expression, which also holds true for the right child "$time$" and "3". Besides, as illustrated in Section 4.1.1, the nodes in formula r carry the concept vectors E_r (for leaf nodes) and semantic embeddings S_r (for non-leaf nodes) with strong semantics, e.g., the semantic embedding of "×" represents "multiplication of $speed$ and $time$". We argue that it can be combined with the correspondence to inject explicit meaning into the reasoning goals, thus ensuring good interpretability.

Formally, if we have selected or inherited a formula r that derives symbol y_t by the mechanisms in Section 4.2.2, here we will use the concept vector (or semantic embedding) of y_t's left/right child in r from Knowledge System, denoted as $\boldsymbol{s}^l/\boldsymbol{s}^r$, to guide generate y_t's left/right sub-goal $\boldsymbol{q}^l$ (i.e., $\boldsymbol{q}_{t+1}$)/$\boldsymbol{q}^r$ respectively. Taking $\boldsymbol{q}^l$ for example, we implement our method as Eq. (14), where $\boldsymbol{c}$ is the context vector [49]. $\boldsymbol{q}^r$ is obtained similarly.

$$\begin{gathered} \boldsymbol{q}^l = \boldsymbol{o}^l \odot \boldsymbol{d}^l, \\ \boldsymbol{o}^l = \sigma(\boldsymbol{W}_o \cdot [\boldsymbol{s}^l, q_t, \boldsymbol{c}]),\ \boldsymbol{d}^l = tanh(\boldsymbol{W}_d \cdot [\boldsymbol{s}^l, \boldsymbol{q}_t, \boldsymbol{c}]) \end{gathered} \tag{14}$$

In summary, our FOMAS mainly has three advantages. First, its Knowledge-Reasoning Systems simulate the human cognitive structure of formula mastery, whose main idea is general to be potentially applicable to different reasoning problems (e.g., physical problems) that also require essential formula knowledge. Second, it provides a novel learning manner that masters both semantics and mathematical logic of formulas, which well models the human understanding characteristics (verified in Section 5.3.1). Third, it imitates the pattern of human thought to conduct reasoning under the instruction of math formulas, which duduces more reliable and reasonable results in a transparent way.

5 EXPERIMENTS

5.1 Experimental Setup

5.1.1 Implementation Details. In Knowledge System, the dimension d of concept vectors is 512, and the epoch of pretraining is 100. In Reasoning System, the dimension of hidden vectors is 512. Specifically, α in Eq. (6) is set as 0.5 and 0.05 on Math23K-F and MAWPS-F respectively, and β in Eq. (8) is set to be 0.2. We will discuss their sensitivity in Section 5.2.3. All parameters are initialized uniformly and trained by Adam [23], with other hyperparameters being set following [27]. For dataset preprocessing, we follow the public train(21,162)/valid(1,000)/test(1,000) partition of Math23K for Math23K-F. For MAWPS-F, the models are evaluated with 5-fold cross-validation. All experiments are run on a Linux server with four 2.30GHz Intel Xeon Gold 5218 CPUs and a Tesla V100 GPU[1].

5.1.2 Baselines. We take the following representative and SOTA MWP models, as well as ChatGPT as baselines for comparison.

- **Seq2Seq** [35]: uses a vanilla seq2seq model with attention to translate MWP to equation templates directly.
- **GTS** [49]: adopts a heuristic goal-driven reasoning manner to generates expression trees.
- **Graph2Tree** [54]: incorporates the relationships and order information among quantities into problem understanding.
- **HMS** [29]: develops a hierarchical word-clause-problem relation to better exploit the problem semantics.
- **NS-Solver** [41]: incorporates four auxiliary tasks into training to master symbolic constraints.
- **BERT-Tree** [26]: employs BERT as semantic encoder and derives expressions by the decoder of GTS.
- **SUMC** [46]: designs a M-tree coding to unify the diverse but equivalent expressions represented by binary trees.

[1] Our datasets and codes are available at *https://github.com/Ljyustc/FOMAS.*

Table 3: **Answer Accuracy ($*: p < 0.05$ w.r.t. BERT-Tree).**

	Math23K-F	MAWPS-F
Seq2Seq	0.640	0.797
GTS	0.756	0.826
Graph2Tree	0.774	0.837
HMS	0.761	0.803
NS-Solver	0.757	/
BERT-Tree	0.833	0.872
SUMC	0.825	0.820
LogicSolver	0.834	/
ChatGPT	0.649	0.883
FOMAS	**0.848***	**0.886***

Table 4: Results of ablation study.

		Math23K-F	MAWPS-F
FOMAS		**0.848**	**0.886**
Knowledge System	w/o legality	0.829	0.875
	w/o flexibility	0.832	0.875
Reasoning System	w/o select	0.839	0.878
	w/o inherit	0.843	0.880
	w/o formula-goal	0.842	0.882

- **LogicSolver** [50]: retrieves logical formulas as prompts to improve problem representations and predicts formulas after expression generation as explanations.
- **ChatGPT**[2]: is a flourishing model trained on large-scale dialogue datasets, which is capable to interact in a conversational way. We input the math word problem into its chatbox and manually extract the numeric answer from its response.

5.2 Performance on Answer Reasoning

5.2.1 Answer Accuracy. Table 3 reports the answer accuracy of all models, and we find several key observations. First, our FOMAS outperforms all the baselines, and by applying paired t-test, its improvements over the SOTA BERT-Tree on both datasets are statistically significant with $p < 0.05$ (marked with $*$). This result demonstrates that mastering the formula knowledge is necessary and valuable to achieve stronger mathematical reasoning ability. Second, FOMAS performs better than LogicSolver which takes formulas as prompts to enrich problem understanding. It reflects that conducting reasoning under the guidance of formulas is necessary and more in line with human cognitive process, and verifies the effectiveness of our proposed formula application mechanisms in Reasoning System. Third, FOMAS obtains comparable performance to ChatGPT on MAWPS-F but is significantly better than it on Math23K-F. As Math23K-F is more difficult than MAWPS-F (i.e., has longer expression that requires more reasoning steps as shown in Table 2), we can conclude that the capability to acquire and explicitly apply formula knowledge makes our FOMAS more robust in figuring out complex mathematical problems (more analyses of ChatGPT are presented in *Appendix B*).

5.2.2 Ablation Study. To examine each part of FOMAS, we conduct the ablation study in Table 4. Specifically, in the Knowledge System, we introduce "w/o legality" and "w/o flexibility" that omit the legality objective L_{leg} and flexibility objective L_{fle} in Eq. (4), respectively. In the Reasoning System, we introduce "w/o select" and "w/o inherit" that remove the formula-selected mechanism and formula-inherited mechanism in Eq. (13), respectively. Moreover, "w/o formula-goal" replaces our formula-guided goal generation in Eq. (14) by the original generation method of GTS [49].

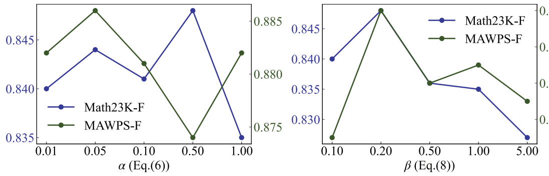


Figure 5: Accuracy with different hyperparameters α and β.

We conclude the results as follows. First, all components of FOMAS contribute to correctly mastering the formula knowledge, because removing each of them leads to a performance decrease. Second, the removal of legality or flexibility has the greatest impact on the effect, which implies that grasping the mathematical logic behind formulas is the foundation of using them. Besides, the two pretraining objectives are almost equally important as there is no significant difference between their results. Third, "w/o select" and "w/o inherit" diminish the results more greatly than "w/o formula-goal", which suggests that formula-guided symbol prediction contributes more to FOMAS than goal generation.

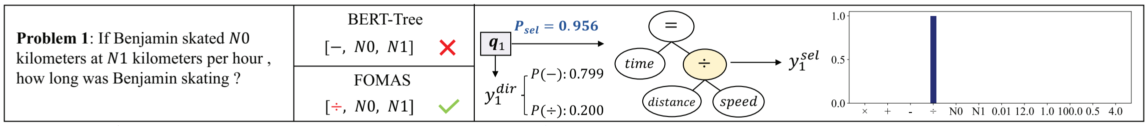
5.2.3 Hyperparameter Sensitivity. In FOMAS, α in Eq. (6) and β in Eq. (8) play an important role for modeling. Specifically, α balances the weight of symbol prediction and formula selection when training FOMAS. β controls the preference to focus more on the reasoning goal or the formula itself when selecting the most appropriate formula for reasoning steps. Figure 5 shows the performances of $\alpha \in \{0.01, 0.05, 0.1, 0.5, 1\}$ and $\beta \in \{0.1, 0.2, 0.5, 1, 5\}$.

As α increases, the accuracy first increases, but then decreases. It indicates that properly balancing the objectives of symbol prediction and formula selection is beneficial. Besides, the peak of performance is attained when $\alpha = 0.5$ and 0.05 on Math23K-F and MAWPS-F, respectively. This suggests the high correlation between the difficulty of mastering the application of formulas and the difficulty of datasets, where a simple one (i.e., MAWPS-F) is easier for knowledge learning (i.e., requires a smaller regularization α). β shows a similar trend, requiring a precise balance between concerns on the reasoning goal and the formula. In particular, since the result is better when $\beta < 0.5$, it implies that compared with the reasoning goal, the relevance of the formula to the problem sentence should be paid more attention in the selection mechanism.

5.3 Analyses of FOMAS

5.3.1 Formula Learning. For formula learning, we expect FOMAS to comprehensively grasp the meanings of formulas, which we verify by investigating the distribution of different formulas' representations. Specifically, we take the semantic vectors of the root $\boldsymbol{s}_r$ of all formulas and their variants after pretraining Eq. (4), and visualize the 5 most frequent ones (with their variants) by T-SNE in Figure 7. Generally, we observe that formulas from different prototypes (i.e., in different colors) are well separated, while those with the same structure feature (e.g., "$rate = work \div time$" and

[2] https://openai.com/blog/chatgpt/. Please see *Appendix B* for more details.



Problem 1: If Benjamin skated $N0$ kilometers at $N1$ kilometers per hour , how long was Benjamin skating ?

BERT-Tree

[−, $N0$, $N1$] ✗

FOMAS

[÷, $N0$, $N1$] ✓

Figure 6: Case 1 (please see Appendix C for cases 2 and 3). We visualize FOMAS’s formula-selected mechanism at $t = 1$.

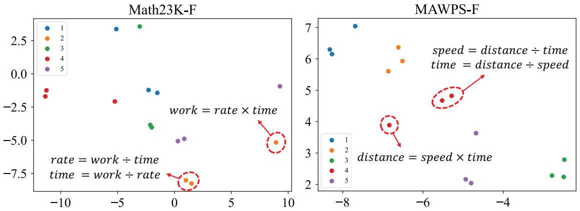


Figure 7: Visualization of the 5 most frequently used formulas (with their variants in the same color) after pretraining Eq. (4). The formula of id 1-5 is referred in Table 1.

Table 5: Performances of formula application.

	Math23K-F		MAWPS-F	
	FOMAS	IP	FOMAS	IP
ACC(↑)	0.954	0.950	0.961	0.957
Precision(↑)	0.749	0.731	0.808	0.776
Recall(↑)	0.687	0.621	0.758	0.742
	FOMAS	BT	FOMAS	BT
PIF(↓)	0.112	0.220	0.134	0.257

“$time = work \div rate$”) are relatively closer. It validates that FOMAS has advantages in formula learning through encoding the structural and lexical information separately. Besides, it also reflects FOMAS’s capability to keep the formula semantics well when conducting mathematical transformation, validating that our pretraining manner has acquired and maintained the knowledge behind it.

5.3.2 Formula Applying. For formula application, one of the core steps of FOMAS is to select the most suitable formula by Eq. (9) in Reasoning System. Thus, we first quantify the performance of FOMAS’ formula-selected mechanism, which covers the correctness of the formula-inherited mechanism. Specifically, we report the *ACC*, *Precision*, and *Recall* from both classification and top-1 ranking aspect and compare the results with “IP” that replaces the $score(r)$ in Eq. (8) by inner product of $\boldsymbol{s}_r$ and $\boldsymbol{q}_t$. From Table 5, our FOMAS achieves the best performances on all metrics, which verifies the effectiveness and robustness of our proposed mechanisms. Second, to more clearly show how FOMAS benefits from mastering the formulas for mathematical reasoning, we compare its solutions with BERT-Tree (“BT”) and calculate the *PIF* metric introduced in Section 3.1. As reported in Table 5, FOMAS significantly reduces this proportion of errors, which explicitly demonstrates that it has learned to use the formula knowledge to conduct more accurate reasoning. After a more detailed inspection of these results, we find that FOMAS performs better at formulas that occur more often, while 31.6% and 63.3% of its mistakes on Math23K-F and MAWPS-F are due to formulas appearing fewer than 30 times in the training set, respectively. Thus, we leave one of the possible future directions is to explore knowledge learning in a few-shot scenario.

5.3.3 Interpretability Verification. Further, we conduct case study to illustrate the interpretable formula application process of FOMAS. We plot one case in Figure 6 and two cases in *Appendix C*. For each case, we first report the problem sentence and the prefix expressions generated by BERT-Tree and FOMAS. Then, we visualize the output symbol distribution of formula-selected/inherited mechanism of FOMAS at steps where it corrects the mistakes of BERT-Tree.

Specifically, for cases 1 and 3, FOMAS correctly reasons “÷” at $t = 1$, while BERT-Tree responds the wrong “−” (“×”) with a high probability 0.799 (0.974). From the visualized formula-selected mechanism at $t = 1$, we observe that FOMAS selects the formula “$time = distance \div speed$” and “$speed = distance \div time$” with $P_{sel} = 0.956$ and 0.999 in cases 1 and 3 respectively, based on which extracts “÷” as y_1^{sel} and corrects the wrong symbol by confidence-based ensemble. This phenomenon confirms the necessity and effectiveness of our formula-selected mechanism and ensemble method. In particular, the selected two formulas are variants of the same “$distance = time \times speed$”. The precise use of them again verifies FOMAS’ mastery of formula semantics and mathematical logic.

For case 2, although BERT-Tree generates the correct first two symbols $y_1 = \times, y_2 = N0$, it still deduces the wrong y_3 as 0.01. Comparatively, FOMAS selects formula “$total\_amount = unit\_amount \times total\_number$” at the first reasoning step $t = 1$. Instructed by this formula, at $t = 3$, it inherits the information of $total\_number$ to calculate “the number of packages” and reasons $y_3^{inh} = N1$ with probability 0.961 by formula-inherited mechanism. With the inherit probability $P_{inh} = 0.997$ in confidence-based ensemble, FOMAS overrides the wrong $P(y_3^{dir} = 0.01) = 0.568$ predicted by direct reasoning (i.e., BERT-Tree) and generates the correct $y_3 = N1$. Thus, formula-inherited mechanism is a crucial and indispensable component of formula knowledge application in FOMAS.

6 CONCLUSION AND FUTURE WORK

In this paper, we focused on the commonsense formula knowledge that is essential for mathematical reasoning. Specifically, we first constructed two benchmark datasets named Math23K-F and MAWPS-F with precise annotations of formula usage to support our study, which are quite general to benefit future research in this field. Then, we proposed a novel Formula-mastered Solver (FOMAS) that contained Knowledge-Reasoning Systems inspired by human cognitive structure and elaborate formula learning/applying mechanisms. Experiments verified FOMAS’ improvements on reasoning accuracy and interpretability, and validated the necessity of formula knowledge for robust reasoning. In the future, we will extend FOMAS to acquire more types of symbolic knowledge from data automatically and generalize to more datasets.

Acknowledgement. This research was partial supported by grants from the National Key Research and Development Program of China (Grant No. 2021YFF0901005) and the National Natural Science Foundation of China (Grants No. 62106244, No. U20A20229).

REFERENCES

[1] Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. MathQA: Towards Interpretable Math Word Problem Solving with Operation-Based Formalisms. In *Proceedings of NAACL-HLT*. 2357–2367.

[2] Yefim Bakman. 2007. Robust understanding of word problems with extraneous information. *arXiv preprint math/0701393* (2007).

[3] Kewei Cheng, Jiahao Liu, Wei Wang, and Yizhou Sun. 2022. RLogic: Recursive Logical Rule Learning from Knowledge Graphs. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 179–189.

[4] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168* (2021).

[5] Alex Davies, Petar Veličković, Lars Buesing, Sam Blackwell, Daniel Zheng, Tomašev, et al. 2021. Advancing mathematics by guiding human intuition with AI. *Nature* 600, 7887 (2021), 70–74.

[6] Robert Benjamin Davis. 1984. *Learning mathematics: The cognitive science approach to mathematics education.* Greenwood Publishing Group.

[7] Michael AE Dummett. 1995. *Frege's philosophy of mathematics.* Harvard University Press.

[8] Jonathan St BT Evans. 2008. Dual-processing accounts of reasoning, judgment, and social cognition. *Annu. Rev. Psychol.* 59 (2008), 255–278.

[9] Jonathan St BT Evans, Stephen E Newstead, and Ruth MJ Byrne. 1993. *Human reasoning: The psychology of deduction.* Psychology Press.

[10] Edward A Feigenbaum, Julian Feldman, et al. 1963. *Computers and thought.* New York McGraw-Hill.

[11] Charles R Fletcher. 1985. Understanding and solving arithmetic word problems: A computer simulation. *Behavior Research Methods, Instruments, & Computers* 17, 5 (1985), 565–571.

[12] Josep Maria Font, Ramon Jansana, and Don Pigozzi. 2003. A survey of abstract algebraic logic. *Studia Logica: An International Journal for Symbolic Logic* 74, 1/2 (2003), 13–97.

[13] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874* (2021).

[14] Marjorie Henningsen and Mary Kay Stein. 1997. Mathematical tasks and student cognition: Classroom-based factors that support and inhibit high-level mathematical thinking and reasoning. *Journal for research in mathematics education* 28, 5 (1997), 524–549.

[15] Zijian Huang, Meng-Fen Chiang, and Wang-Chien Lee. 2022. LinE: Logical Query Reasoning over Hierarchical Knowledge Graphs. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 615–625.

[16] Zhenya Huang, Xin Lin, Hao Wang, Qi Liu, Enhong Chen, Jianhui Ma, Yu Su, and Wei Tong. 2021. Disenqnet: Disentangled representation learning for educational questions. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 696–704.

[17] Zhenya Huang, Qi Liu, Weibo Gao, Jinze Wu, Yu Yin, Hao Wang, and Enhong Chen. 2020. Neural mathematical solver with enhanced formula structure. In *SIGIR*. 1729–1732.

[18] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. 1991. Adaptive mixtures of local experts. *Neural computation* 3, 1 (1991), 79–87.

[19] Zhanming Jie, Jierui Li, and Wei Lu. 2022. Learning to Reason Deductively: Math Word Problem Solving as Complex Relation Extraction. In *ACL*. 5944–5955.

[20] Daniel Kahneman. 2011. *Thinking, fast and slow.* Macmillan.

[21] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT*. 4171–4186.

[22] Hyunju Kim, Junwon Hwang, Taewoo Yoo, and Yun-Gyung Cheong. 2022. Improving a Graph-to-Tree Model for Solving Math Word Problems. In *IMCOM*. 1–7.

[23] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[24] Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. 2016. MAWPS: A math word problem repository. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 1152–1157.

[25] George Lakoff and Rafael Núñez. 2000. *Where mathematics comes from.* Vol. 6. New York: Basic Books.

[26] Zhongli Li, Wenxuan Zhang, Chao Yan, Qingyu Zhou, Chao Li, Hongzhi Liu, and Yunbo Cao. 2022. Seeking Patterns, Not just Memorizing Procedures: Contrastive Learning for Solving Math Word Problems. In *Findings of the Association for Computational Linguistics: ACL 2022*. 2486–2496.

[27] Zhenwen Liang, Jipeng Zhang, Lei Wang, Wei Qin, Yunshi Lan, Jie Shao, and Xiangliang Zhang. 2022. MWP-BERT: Numeracy-augmented pre-training for math word problem solving. In *NAACL-HLT (Findings)*. 997–1009.

[28] Xin Lin, Zhenya Huang, Hongke Zhao, Enhong Chen, Qi Liu, Defu Lian, Xin Li, and Hao Wang. 2023. Learning Relation-Enhanced Hierarchical Solver for Math Word Problems. *IEEE Transactions on Neural Networks and Learning Systems* (2023).

[29] Xin Lin, Zhenya Huang, Hongke Zhao, Enhong Chen, Qi Liu, Hao Wang, and Shijin Wang. 2021. HMS: A Hierarchical Solver with Dependency-Enhanced Understanding for Math Word Problem. In *AAAI*, Vol. 35. 4232–4240.

[30] Zi Lin, Jeremiah Zhe Liu, and Jingbo Shang. 2022. Towards Collaborative Neural-Symbolic Graph Semantic Parsing via Uncertainty. In *Findings of the Association for Computational Linguistics: ACL 2022*. 4160–4173.

[31] Jiayu Liu, Zhenya Huang, Xin Lin, Qi Liu, Jianhui Ma, and Enhong Chen. 2022. A Cognitive Solver with Autonomously Knowledge Learning for Reasoning Mathematical Answers. In *2022 IEEE International Conference on Data Mining (ICDM)*. IEEE, 269–278.

[32] Jiayu Liu, Zhenya Huang, Chengxiang Zhai, and Qi Liu. 2023. Learning by Applying: A General Framework for Mathematical Reasoning via Enhancing Explicit Knowledge Learning. *arXiv preprint arXiv:2302.05717* (2023).

[33] Qi Liu, Zhenya Huang, Yu Yin, Enhong Chen, Hui Xiong, Yu Su, and Guoping Hu. 2021. EKT: Exercise-Aware Knowledge Tracing for Student Performance Prediction. *IEEE Transactions on Knowledge and Data Engineering* 33, 1 (2021), 100–115.

[34] Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-chun Zhu. 2021. Inter-GPS: Interpretable Geometry Problem Solving with Formal Language and Symbolic Reasoning. In *ACL/IJCNLP*. 6774–6786.

[35] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 1412–1421.

[36] Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B Tenenbaum, and Jiajun Wu. 2019. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. In *ICLR*.

[37] Arindam Mitra and Chitta Baral. 2016. Learning to use formulas to solve simple arithmetic problems. In *ACL*. 2144–2153.

[38] Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP Models really able to Solve Simple Math Word Problems?. In *NAACL-HLT*. 2080–2094.

[39] Shuai Peng, Di Fu, Yong Cao, Yijun Liang, Gu Xu, Liangcai Gao, and Zhi Tang. 2022. Compute Like Humans: Interpretable Step-by-step Symbolic Computation with Deep Neural Network. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1348–1357.

[40] Shuai Peng, Ke Yuan, Liangcai Gao, and Zhi Tang. 2021. Mathbert: A pre-trained model for mathematical formula understanding. *arXiv preprint arXiv:2105.00377* (2021).

[41] Jinghui Qin, Xiaodan Liang, Yining Hong, Jianheng Tang, and Liang Lin. 2021. Neural-Symbolic Solver for Math Word Problems with Auxiliary Tasks. In *ACL/IJCNLP*. 5870–5881.

[42] Meng Qu and Jian Tang. 2019. Probabilistic logic neural networks for reasoning. *Advances in neural information processing systems* 32 (2019).

[43] Shuming Shi, Yuehui Wang, Chin-Yew Lin, Xiaojiang Liu, and Yong Rui. 2015. Automatically solving number word problems by semantic parsing and reasoning. In *EMNLP*. 1132–1142.

[44] Steven A Sloman, Richard Patterson, and Aron K Barbey. 2021. Cognitive neuroscience meets the community of knowledge. *Frontiers in Systems Neuroscience* (2021), 120.

[45] Matthias Steup and Ram Neta. 2005. Epistemology. (2005).

[46] Bin Wang, Jiangzhou Ju, Yang Fan, Xin-Yu Dai, Shujian Huang, and Jiajun Chen. 2022. Structure-Unified M-Tree Coding Solver for MathWord Problem. *EMNLP*.

[47] Yan Wang, Xiaojiang Liu, and Shuming Shi. 2017. Deep neural solver for math word problems. In *EMNLP*. 845–854.

[48] Qinzhuo Wu, Qi Zhang, Jinlan Fu, and Xuan-Jing Huang. 2020. A knowledge-aware sequence-to-tree network for math word problem solving. In *EMNLP*. 7137–7146.

[49] Zhipeng Xie and Shichao Sun. 2019. A Goal-Driven Tree-Structured Neural Model for Math Word Problems.. In *IJCAI*. 5299–5305.

[50] Zhicheng Yang, Jinghui Qin, Jiaqi Chen, Liang Lin, and Xiaodan Liang. 2022. LogicSolver: Towards Interpretable Math Word Problem Solving with Logical Prompt-enhanced Learning. In *Findings of EMNLP*.

[51] Dongran Yu, Bo Yang, Dayou Liu, and Hui Wang. 2021. A Survey on Neural-symbolic Systems. *arXiv preprint arXiv:2111.08164* (2021).

[52] Dongran Yu, Bo Yang, Qianhao Wei, Anchen Li, and Shirui Pan. 2022. A Probabilistic Graphical Model Based on Neural-Symbolic Reasoning for Visual Relationship Detection. In *CVPR*. 10609–10618.

[53] Dongxiang Zhang, Lei Wang, et al. 2020. The Gap of Semantic Parsing: A Survey on Automatic Math Word Problem Solvers. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42, 9 (2020), 2287–2305.

[54] Jipeng Zhang, Lei Wang, Roy Ka-Wei Lee, et al. 2020. Graph-to-Tree Learning for Solving Math Word Problems. In *ACL*. 3928–3937.

[55] Hongke Zhao, Chuang Zhao, Xi Zhang, Nanlin Liu, Hengshu Zhu, Qi Liu, and Hui Xiong. 2023. An Ensemble Learning Approach with Gradient Resampling for Class-Imbalance Problems. *INFORMS Journal on Computing* (2023).

[56] Wayne Xin Zhao, Kun Zhou, Zheng Gong, et al. 2022. JiuZhang: A Chinese Pre-trained Language Model for Mathematical Problem Understanding. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4571–4581.

Table 6: All 51 math formulas on Math23K-F.

$distance = speed \times time$
$work = rate \times time$
$consume = consume\_rate \times time$
$total\_time = unit\_time \times total\_number$
$total\_weight = unit\_weight \times total\_number$
$total\_cost = unit\_cost \times total\_number$
$total\_volume = unit\_volume \times total\_number$
$total\_income = unit\_income \times total\_number$
$total\_amount = unit\_amount \times total\_number$
$toatl\_area = unit\_area \times total\_number$
$total\_length = unit\_length \times total\_number$
$rectangle\_area = length \times width$
$rectangle\_perimeter = (length + width) \times 2$
$square\_area = side \times side$
$square\_perimeter = side \times 4$
$circle\_perimeter = 2 \times \pi \times radius$
$circle\_perimeter = \pi \times diameter$
$circle\_area = \pi \times radius \wedge 2$
$circle\_area = \pi \times radius \times radius$
$circle\_area = \pi \times (diameter \div 2) \wedge 2 \times radius$
$parallelogram\_area = length \times height$
$triangle\_area = length \times height \div 2$
$trapezium\_area = (side1 + side2) \times height \div 2$
$trapezium\_area = plus\_side1\_side2 \times height \div 2$
$cubiod\_volume = base\_area \times height$
$cubiod\_volume = length \times width \times height$
$cube\_volume = side \times side \times side$
$cone\_volume = base\_area \times height \div 3$
$cylinder\_area = 2 \times \pi \times radius \times height$
$cylinder\_area = \pi \times diameter \times height$
$cylinder\_area = base\_perimeter \times height$
$cylinder\_volumne = radius \wedge 2 \times \pi \times height$
$cylinder\_volumne = (diameter \div 2) \wedge 2 \times \pi \times height$
$cylinder\_volume = base\_area \times height$
$cylinder\_volume = \pi \times radius \times radius \times height$
$interest = principle \times rate \times time$
$income = savings + expenditure$
$sell\_price = profit + cost\_price$
$sell\_price = cost\_price \times (1 + profit\%)$
$sell\_price = cost\_price + cost\_price \times profit\%$
$tax\_amount = tax\_income \times tax\_rate$
$tax\_income = salary - base$
$actual\_salary = salary - tax\_amount$
$average = sum\_of\_terms \div number\_of\_terms$
$probability = number\_of\_desired \div number\_possibility$
$percent\_change\% = (final - initial) \div initial$
$last\_term = first\_term + (n - 1) \times common\_difference$
$weight = density \times volume$
$weight\_substance = weight\_all \times concentration\%$
$2nd\_number = lcm \times hcf \div 1st\_number$
$reciprocal = 1 \div number$

A MORE DATASET ANALYSES

Tables 6 and 7 list all math formulas on our benchmark datasets, and the distributions of their usage frequency are visualized in Figure 8. Besides, we count the number of used formulas for each problem and report the distribution in Table 8. It indicates that 33.5% and 38.4% problems require at least one formula on Math23K-F and MAWPS-F, respectively, as we have reported in Section 3.1. Especially, those requiring one formula (i.e., 4, 520 and 860) account for 19.5% and 36.2% on each dataset. This result verifies that formulas are very important for reasoning expressions correctly, thus validating the motivation and contribution of our work.

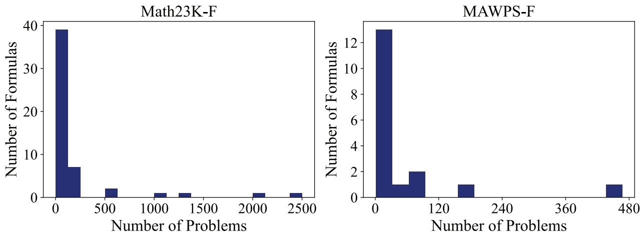


Figure 8: Distributions of formula usage frequency.

Table 7: All 18 math formulas on MAWPS-F.

$distance = speed \times time$
$work = rate \times time$
$consume = consume\_rate \times time$
$total\_time = unit\_time \times total\_number$
$total\_weight = unit\_weight \times total\_number$
$total\_cost = unit\_cost \times total\_number$
$total\_volume = unit\_volume \times total\_number$
$total\_income = unit\_income \times total\_number$
$total\_amount = unit\_amount \times total\_number$
$rectangle\_area = length \times width$
$interest = principle \times rate \times time$
$income = savings + expenditure$
$sell\_price = profit + cost\_price$
$sell\_price = cost\_price \times (1 + profit\%)$
$average = sum\_of\_terms \div number\_of\_terms$
$probability = number\_of\_desired \div number\_possibility$
$percent\_change = (final - initial) \div initial \times 100$
$last\_term = first\_term + (n - 1) \times common\_difference$

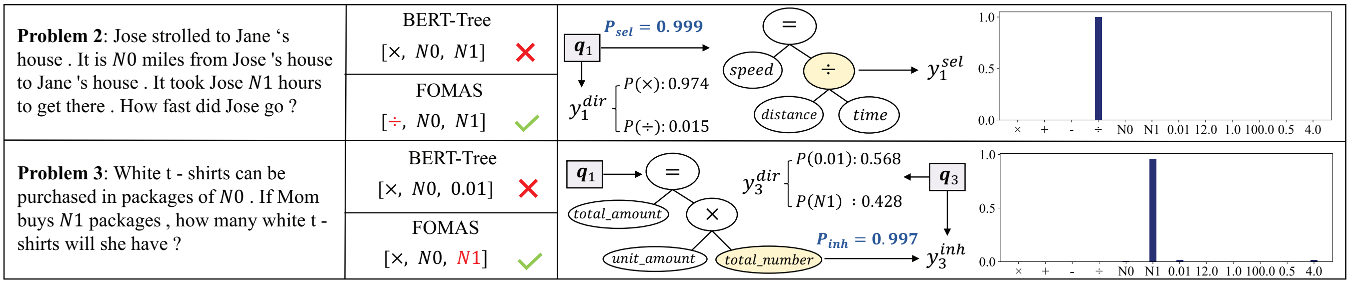
Table 8: Distributions of the number of used formulas.

# Used Formulas	# Problems (Math23K-F)	# Problems (MAWPS-F)
0	14,412	1,462
1	4,520	860
2	3,005	33
More than 2	225	18

B CHATGPT: EXPERIMENTS AND ANALYSES

ChatGPT is a recently released AI chatbot developed by OpenAI (https://openai.com/), which has garnered significant attention due to its strong language generation capability. Equipped with a vast array of factual knowledge, it is capable of responding problems across domains such as history, culture, and science.

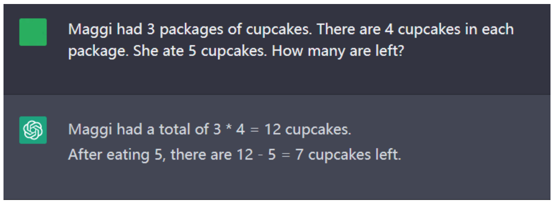
Here, we are interested in evaluating ChatGPT's performance in answering math word problems. We conducted this experiment in January, 2023, and OpenAI did not publish the code API at that time. Therefore, we can only access the official website tool (https://openai.com/blog/chatgpt/) to obtain its responses to all problems (by a crawler), and then calculate the answers. Specifically, we show an example in Figure 10. The problem ("Maggi had ... left?") is input into its online chatbox without any restrictions or



Problem 2: Jose strolled to Jane 's house . It is $N0$ miles from Jose 's house to Jane 's house . It took Jose $N1$ hours to get there . How fast did Jose go ?

Problem 3: White t - shirts can be purchased in packages of $N0$. If Mom buys $N1$ packages , how many white t - shirts will she have ?

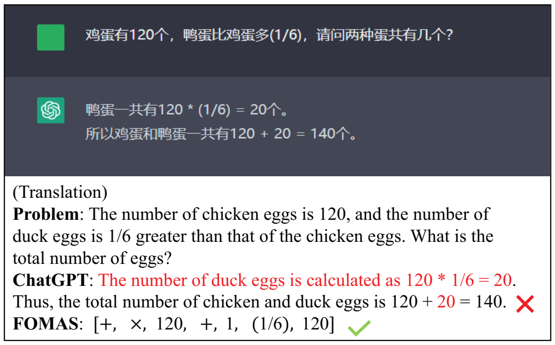
Figure 9: Cases 2 and 3. For case 2, we visualize FOMAS's formula-inherited mechanism at $t = 3$. For case 3, we visualize the formula-selected mechanism at $t = 1$.



Maggi had 3 packages of cupcakes. There are 4 cupcakes in each package. She ate 5 cupcakes. How many are left?

Maggi had a total of 3 * 4 = 12 cupcakes.
After eating 5, there are 12 - 5 = 7 cupcakes left.

Figure 10: Example of ChatGPT's response.

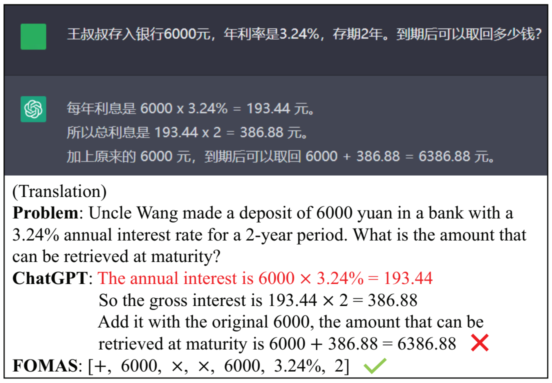


鸡蛋有120个，鸭蛋比鸡蛋多(1/6)，请问两种蛋共有几个?

鸭蛋一共有120 * (1/6) = 20个。
所以鸡蛋和鸭蛋一共有120 + 20 = 140个。

(Translation)
Problem: The number of chicken eggs is 120, and the number of duck eggs is 1/6 greater than that of the chicken eggs. What is the total number of eggs?
ChatGPT: The number of duck eggs is calculated as 120 * 1/6 = 20.
Thus, the total number of chicken and duck eggs is 120 + 20 = 140. ✗
FOMAS: [+, ×, 120, +, 1, (1/6), 120] ✓

Figure 11: A fraction case between ChatGPT and FOMAS.



王叔叔存入银行6000元，年利率是3.24%，存期2年。到期后可以取回多少钱?

每年利息是 6000 x 3.24% = 193.44 元。
所以总利息是 193.44 x 2 = 386.88 元。
加上原来的 6000 元，到期后可以取回 6000 + 386.88 = 6386.88 元。

(Translation)
Problem: Uncle Wang made a deposit of 6000 yuan in a bank with a 3.24% annual interest rate for a 2-year period. What is the amount that can be retrieved at maturity?
ChatGPT: The annual interest is 6000 × 3.24% = 193.44
So the gross interest is 193.44 × 2 = 386.88
Add it with the original 6000, the amount that can be retrieved at maturity is 6000 + 386.88 = 6386.88 ✗
FOMAS: [+, 6000, ×, ×, 6000, 3.24%, 2] ✓

Figure 12: An arithemetic case between ChatGPT and FOMAS.

illustrations. We scrape the whole response ("Maggi had a totoal ... 7 cupcakes left.") and manually extract the numeric answer (i.e., 7) as ChatGPT's result. Moreover, since the responses of ChatGPT may be different with multiple trials, we operate the above processes three times to obtain the stable results for all problems.

As reported in Table 3, ChatGPT obtains answer accuracy of 64.9% and 88.3% on Math23K-F and MAWPS-F respectively, which reflects that it has basic mathematical reasoning ability. Specially, the performance on Math23K-F indicates that ChatGPT may still lack sufficient Chinese MWP corpus, and has some room for improvement on the problems that require more reasoning steps (recall that Math23K-F is a Chinese benchmark dataset that contains more difficult problems than MAWPS-F through the analysis in Section 3.1). We also make case study on Math23K-F, and find that there are mainly two typical errors that should be concerned. First, it struggles to answer the problems that examine fraction or proportion as plotted in Figure 11. Second, its arithmetic capability remains to be further improved. It can derive the correct expression for a problem, but makes a mistake when calculating the final numeric answer, e.g., in Figure 12, it gets the wrong value 193.44 based on the correct expression 6000 × 3.24% (marked red).

Besides, for a more sufficient comparison, we design another baseline named ChatGPT-P by prompting all our annotated formulas together with the problem sentence as the input of ChatGPT. Our prompt is: "[problem sentence]. You may use the following math formulas to solve this problem: [formulas]", and we also manually extract the numeric answer from its response. ChatGPT-P obtains 58.4% and 89.2% answer accuracy on Math23K-F and MAWPS-F, respectively. After suspecting its outputs, we found that on Math23K-F, ChatGPT-P might misunderstand the task after giving all the formulas. For example, it might just type out all formulas and output "\n $speed = distance \div time$, \n $rate = work \div time$, ..." instead of actually solving the problem. We guess it is probably because the 51 formulas (reported in Table 2) in the dataset take up most of the place in the prompt and make ChatGPT frustrated with what to do, thus resulting in a performance drop. While for MAWPS-F that has fewer (i.e., 18) formulas, prompting can improve the answer accuracy, which shows the necessity of studying formula knowledge on mathematical reasoning in this paper, and verifies the validity of our manually constructed formulas from another perspective. The different performances above inspire us to explore how to combine large-scale language models with external knowledge without introducing too much noise. We think it will be an interesting and valuable research direction in the future.

C CASES 2 AND 3

We show one case in Section 5.3.3 and another two cases in Figure 9. Pleaser refer to Section 5.3.3 for detailed explanations and analyses.