Pattern Recognition 171 (2026) 112261

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Reasoning step by step via a neural-symbolic geometry problem solver

Yaxian Wang [a,b], Bifan Wei [b,c,*], Yinghong Ma [b], Xudong Jiang [d], Henghui Ding [e], Zhongmin Cai [f], Jun Liu [b]

[a] School of Information Engineering, Chang'an University, Xi'an, Shaanxi, 710064, China
[b] School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an, Shaanxi, 710049, China
[c] School of Continuing Education, Xi'an Jiaotong University, Xi'an, 710049, China
[d] School of Electrical and Electronic Engineering, Nanyang Technological University, 639798, Singapore
[e] Institute of Big Data, Fudan University, Shanghai, 200438, China
[f] Institute of System Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi, 710049, China

ARTICLE INFO

Keywords:
Geometry problem solving
Diagram understanding
Multimodal representation
Neural-symbolic reasoning
Interpretable reasoning

ABSTRACT

Geometry problem solving (GPS) is a challenging task that requires comprehensive multimodal geometry representation and reasoning capabilities. Existing approaches still have limitations, primarily due to their insufficient understanding of multimodal geometry context and their heavy reliance on costly solution programs for interpretability. Therefore, we propose a Neural-Symbolic Geometry Problem Solver (NS-GPS), which performs step-by-step reasoning to generate the intermediate solution programs with the answers as guidance. Drawing upon the dual process theory in cognitive science, NS-GPS integrates a progressive fusion module for robust multimodal semantic understanding of the geometry diagram and text problem, as well as a neural-symbolic reasoning module for step-wise sub-program generation and execution. Additionally, the intermediate executed results of the sub-programs provide training feedback by comparing with the answer to promote solution program generation. Extensive experiments on four datasets demonstrate that NS-GPS achieves competitive performance to the state-of-the-arts, while achieving interpretability with less cost.

1. Introduction

Geometry problem solving (GPS) as a challenging vision-language task has recently attracted extensive research attention [1–5]. GPS task serves as a good test-bed to evaluate the multimodal geometry understanding and reasoning ability of machines, which has important research values and application potential for the intelligent education field. A typical geometry problem consists of a geometry diagram and a text problem as shown in Fig. 1. The geometry diagram presents the structural information between the geometry primitives such as the points and lines. The text problem generally describes the numerical information (*e.g.*, $\angle A = 125°$) and semantic information (*e.g.*, $AD \mathbin{//} BC$) about the geometry primitives and puts forward a question as the reasoning goal at last such as "*the degree of* $\angle BCE$ *is ( )*". To answer such a geometry problem correctly, the machine requires two key abilities: (*i*) multimodal representation ability about the geometry context including the geometry diagram and text problem; (*ii*) geometry reasoning ability based on the multimodal context and geometric knowledge.

Researchers have recently made great efforts on GPS task [2–5] from the above two aspects. **First**, for the multimodal representation of the geometry diagram and text problem, existing methods [2,3] encode the text problem as a sequence of words and then model their interaction with the geometry diagram. However, compared with the individual words, each clause generally expresses more complete semantics, which describes the information about one geometry primitive or primitive relation. Rather than processing all words in the text problem together to model the semantics of the entire sentence, humans usually read it part-by-part (*e.g.*, the clause) and match the local details to the relevant geometry primitives in the diagram. Hence, these methods are far from enough for effective multimodal semantic understanding about the geometry context, leading to undesirable behaviors for guiding the subsequent geometry reasoning.

Second, for geometry reasoning, some pioneering work focuses on making geometry problem solving interpretable by extracting solution programs as the explicit reasoning process based on multimodal context [2–4]. These methods follow a generation-then-execution paradigm, which aims to generate a complete solution program based on a seq2seq framework [6] and then execute the program to obtain the final answer. However, the solving paradigm assumes the availability of exhaustive intermediate solution program annotations besides answers for train-

* Corresponding author.
E-mail addresses: wyx1566@chd.edu.cn (Y. Wang), weibifan@xjtu.edu.cn (B. Wei).

https://doi.org/10.1016/j.patcog.2025.112261
Received 7 May 2024; Received in revised form 10 June 2025; Accepted 26 July 2025
Available online 11 August 2025

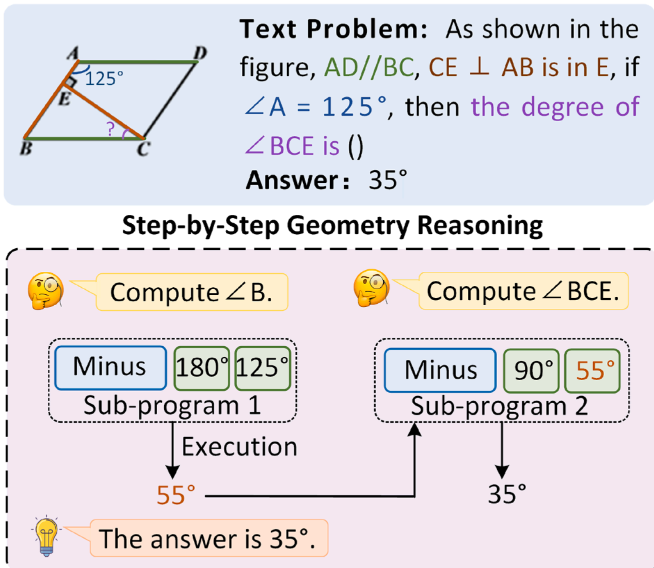

Fig. 1. One example of a geometry problem, and the step-by-step geometry reasoning process consisting of neural sub-program generation and symbolic sub-program execution for numerical computation.

ing. Instead of directly optimizing the answer accuracy, the question-answering task is formulated as a sequence generation task to generate a complete solution program sequence with the annotated one program as the supervision. Although the generated sequential solution program can provide interpretability, it is inconsistent with a step-wise problem-solving process similar to humans. It is also more expensive due to the time-consuming and labor-intensive annotation cost compared to acquiring readily available answers. Moreover, unlike the unique result, one geometry problem may have diverse solutions to reach the answer. Hence, fitting one solution with such unique strong supervision may limit the flexibility of models.

To address these issues, we propose a **N**eural-**S**ymbolic **G**eometry **P**roblem **S**olver (NS-GPS) to perform geometry reasoning step by step, which generates an intermediate solution program with the answers as the guidance. Drawing upon the dual process theory [7,8] in cognitive science, NS-GPS consists of a multimodal representation encoder to extract a comprehensive multimodal representation of the geometry diagram and text problem, and a neural-symbolic reasoning decoder to generate the intermediate solution program and the final answer. For the multimodal representation encoder, inspired by the human cognition habit, we propose a Progressive Fusion Module (PFM) to model the relationships between the structural geometry diagram and the fine-grained word-level and semantic-complete clause-level information about the geometry primitives in the text problem. PFM can help better understand the contextual semantic details from local to global and multi-modal perspectives. After that, based on the comprehensive multimodal representation, we construct a novel neural-symbolic reasoning decoder to perform step-by-step geometry reasoning consisting of neural sub-program generation and symbolic sub-program execution for numerical computation, as illustrated in Fig. 1. At each step, a sub-program is generated by an operator selector and operand selector, which is then executed via a sub-program executor to obtain a result. Additionally, by decomposing the problem solving process step by step, the intermediate executed result acts as a feedback signal via matching the final answer, further promoting the sub-program generation toward the final answer. The intermediate result is stored in a dynamic operand dictionary in real time for subsequent sub-program generation. The proposed NS-GPS is trained end-to-end with the discrete rewards derived from matching the ground truth and the predicted answer in a reinforcement learning framework with the answer as the guidance. Without providing explicit solutions in the training process, the model is encouraged to perform step-by-step reasoning to generate them by maximizing the reward signal from the answer. Therefore, NS-GPS is not simply imitating pre-existing solutions but actively engages in step-wise reasoning to construct intermediate steps leading to the correct answer. In summary, our proposed NS-GPS provides both the intermediate solution program and the final answer, achieving interpretability for geometry problem solving with significantly reduced cost.

Our main contributions can be summarized as follows:

- We propose a Neural-Symbolic Geometry Problem Solver (NS-GPS) to perform geometry reasoning step by step, which builds a bridge between the solution programs and answers with the answers as the guidance, achieving interpretability with less annotation cost.
- We design a progressive fusion module to enhance the multimodal geometry context understanding by incorporating the abstract geometry diagram with the fine-grained word-level and semantic-complete clause-level information about the geometry primitives in the text problem.
- We present a neural-symbolic reasoning decoder to perform step-wise geometry reasoning including sub-program generation and execution, which is trained in a reinforcement learning framework using discrete rewards from the answer matching.
- Extensive experiments are conducted on the four benchmark datasets of GeoQA, GeoQA+, Geometry3K, and PGPS9K to demonstrate the advantages of our proposed NS-GPS over existing state-of-the-art methods.

The rest of the paper is organized as follows. Section 2 briefly introduces existing related work. Section 3 describes the details of NS-GPS. Section 4 presents the experimental details and analysis of the results for our proposed method. Section 5 concludes our work and puts forward future work.

2. Related work

2.1. Diagram understanding

In recent years, there has been a surge of research and development in the area of diagram understanding. There are different kinds of diagrams, such as charts [9] for statistics, illustrative diagrams [10] in science textbooks (*e.g.*, geography, biology, and physics) for conveying scientific knowledge, and geometry diagrams [2] for expressing geometry knowledge. These kinds of diagrams are typically presented as simple shapes and color blocks, omitting unnecessary background or texture information. However, they can convey rich and complex knowledge that would typically require multiple sentences to describe. Our work focuses on geometry diagrams. The geometry diagrams embody rich structural and semantic knowledge about geometry primitives (*e.g.*, point, line, and circle) and geometry relationships (*e.g.*, perpendicular and parallel). Effective diagram understanding can promote the performance of downstream tasks such as geometry problem solving. Zhang et al. [11] focused on diagram parsing, which aims to extract the geometry primitives and reason about the geometry relationships. Lu et al. [5] proposed an automatic diagram parser to detect geometry primitives and symbols, and then transform them into a relation set defined in formal language. Peng et al. [12] parsed the diagram and text into a geometry logic graph to encode the geometric information. Chen et al. [2] proposed two auxiliary tasks, including jigsaw location prediction and geometry elements prediction, to enhance diagram semantic representation. It should be noted that the geometry diagrams in both GeoQA [2] and GeoQA+ [3] have limited annotations, which makes it challenging for existing diagram parsers to understand them at a fine-grained level. Therefore, following [2,3], we focus on improving the diagram's semantic representation by exploiting textual semantics fully.

2.2. Multimodal reasoning

Multimodal reasoning has attracted increasing attention due to the development of visual-language tasks such as visual question answering [13–15], diagram question answering [10,16,17], and chart question answering [9,18]. These tasks all require multimodal reasoning over the text problem and images to arrive at the final answer. Because of the different types of images, the reasoning methods also differ slightly. Taking the popular visual question answering task about natural images as an example, recent years have seen a proliferation of datasets and approaches. There are two main lines to achieve visual question answering. The first achieves a good multimodal joint representation through implicit reasoning, such as the joint embedding method [13,19], and attention-based methods [14,20]. This type of work is less interpretable. The second line of work addresses the QA task through explicit reasoning, which translates the question into executable, symbolic programs, achieving a higher level of interpretability. The neural-symbolic methodology has been employed to model reasoning abilities in visual question answering. Specifically, Neural Module Networks [21–23] are proposed as multi-step models that build question-specific layouts and execute them over an image, which provide explicit programs as sequences of symbolic tokens. A domain-specific language (DSL) [23] is adopted to translate a text question into an executable program, which covers a set of fundamental operations for visual reasoning, such as filtering out objects with certain concepts, querying the attribute of an object, or counting the objects with specific concepts. However, the DSL is mainly designed for the specialized CLEVR [24] dataset, leading to the difficulty in generalizing to other datasets. This type of work is similar to ours. Nevertheless, our GPS pays attention to the different kinds of geometry diagrams, which need to perform complex logical reasoning over numerical information combined with multimodal context and geometric theorem knowledge. Therefore, it is essential to design unique multimodal reasoning methods suitable for the special geometry problem solving task.

2.3. Geometry problem solving

A few datasets and methods have recently been proposed for geometry problem solving. Seo et al. [1] proposed the first fully-automated geometry problems solver GEOS, which generates a set of relations about the question text and diagram, and then selects a subset of the relations most related to the joint text and diagram for the final answer. However, it reduces the task as an optimization problem to select the answer that satisfies all constraints and also highly relies on handcrafted rules. Based on the GEOS, Sachan et al. [25] proposed to replace the numerical solver of GEOS with an axiomatic solver, which performs more interpretable logical inference with these horn clause rules and the formal problem description. The above methods are verified on the datasets with a small scale. Later on, Lu et al. [5] first constructed a large-scale dataset Geometry3K to promote the research of geometry problem solving and proposed Inter-GPS to provide the theorem application sequence as explicit reasoning processes based on the unified formal language about the diagram and text problem. However, the Inter-GPS is also a rule-based method and hence hard to extend. Moreover, the theorem search process is slow due to many unnecessary steps and a large search space, and it also does not align with human problem solving thinking. To promote the interpretability research of geometry problem solving, GeoQA [2] and GeoQA+ [3] datasets are proposed with the exhaustive dense annotation of solution programs. The solvers utilize a seq2seq framework to embed a multimodal representation of the geometry diagram and text problem and then generate a solution program. NGS [2] and SCA-GPS [26] propose self-supervised auxiliary tasks to enhance the diagram representation. SCA-GPS [26] also introduces a Character-Diagram Feature Alignment module, which aligns symbolic character features with geometry diagrams to enable the extraction of diagram-aware character representations. Then, it is mainly dedicated to enhancing textual representations by merging character representations from the alignment module into unified semantic information for consecutive character subsequences. In contrast, our method primarily aims to enhance geometry diagram understanding, which fuses word- and clause-level textual representations with diagram representations progressively to enable a robust text-aware diagram representation. DPE-NGS [3] improves NGS from the text representation ability by combining RoBERTa and Bi-LSTM encoders. DualGeoSolver [27] constructs a knowledge-inference system to retrieve relevant geometric knowledge from an external knowledge base and incorporate it into the reasoning process to promote the solution program generation. In contrast to GeoQA and GeoQA+, PGPSNet [28] proposes the PGPS9K dataset with both fine-grained diagram annotation and solution programs. PGPSNet uses textual clauses to express the structural and semantic information in geometry diagrams and adopts structural and semantic pre-training to fuse multi-modal information. LANS [29] proposes layout-aware fusion attention to enhance the intra-modal and cross-modal token fusion between diagram and text, further enhancing diagram layout awareness. However, it remains constrained to point primitives for layout understanding. Ma et al. [30] proposed an improved DenseNet network incorporating two auxiliary tasks, diagram position prediction and geometric element prediction, to enhance cross-modal semantic representation for better geometric element extraction and character recognition. While existing solvers have achieved impressive performance, they are still short in comprehensive multimodal semantic interaction between text problems and geometry diagrams. Moreover, although this kind of solver provides human-readable solving steps, it relies heavily on the full intermediate solution programs as the supervision in the training process. Massive annotated solution programs are expensive and unavailable in real life compared with the answers, thus restricting their application. Therefore, it is necessary to balance interpretability and cost to help expand the application of the geometry problem solving task.

3. Approach

In this section, we first formalize the task of Geometry Problem Solving (GPS) in Section 3.1. Then, we describe the details of our proposed approach, Neural-Symbolic Geometry Problem Solver (NS-GPS). The overall framework of NS-GPS is illustrated in Fig. 2. Apparently, NS-GPS contains two primary parts: multimodal representation encoder and neural-symbolic reasoning decoder, introduced in Sections 3.2 and 3.3 respectively.

3.1. Task formalization

Given the multimodal context including a geometry diagram D, and a text problem T, geometry problem solving aims to generate an intermediate solution program $P = \{op_t, [od_t^i]_{i=0}^N\}_{t=0}^M$ and a final answer A. Specifically, the intermediate solution program contains M sub-programs, where each sub-program consists of an operator op_t and several operands od_t^i. By generating and executing these sub-programs step by step, the final answer A can be derived gradually.

3.2. Multimodal representation encoder

The geometry diagram displays the structural appearance information, *e.g.*, *CD is a line, and point E lies on the line AB*. In the text problem, higher-level abstraction of the diagram information is provided, which generally includes several clauses to describe the numerical information for the geometry primitives (*e.g.*, $\angle A = 125°$) or the semantic relations between the geometry primitives (*e.g.*, $CE \perp AB$, $AD \,//\, BC$). Each clause is simple but expresses complete semantics, which describes information about one geometric primitive or primitive relation. The geometry diagrams express rich and complex structural and semantic information, and even the advanced large multimodal models [31,32] struggle with effective geometry diagram understanding without the aid of the text.

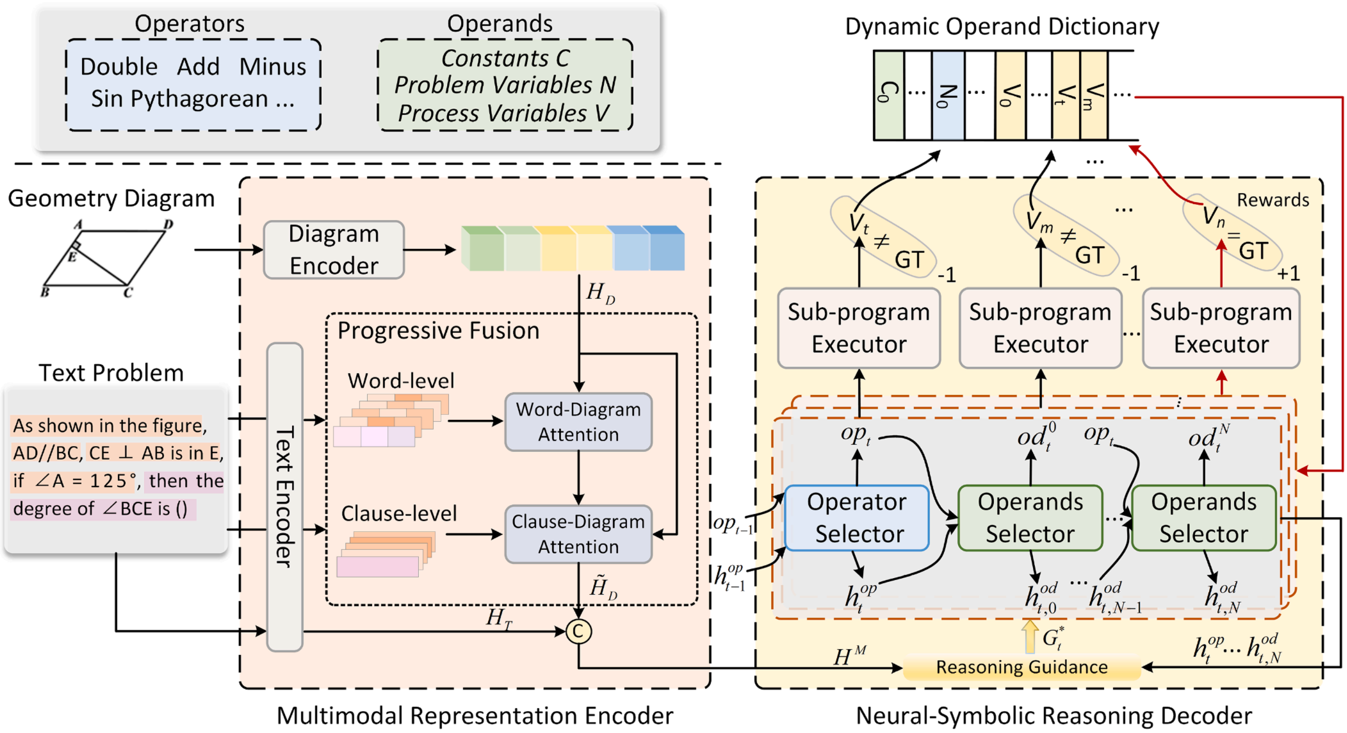

Fig. 2. The overall framework of the proposed NS-GPS consisting of a multimodal representation encoder and a neural-symbolic reasoning decoder. (1) the multimodal representation encoder takes a geometry diagram D and a text problem T as input, and adopts a progressive fusion module to generate a semantic-enhanced diagram representation $\widetilde{H}_D$. The comprehensive multimodal context representation H^M is used to guide the subsequent geometry reasoning. (2) the neural-symbolic reasoning decoder mainly consists of reasoning guidance, operator selector, operand selector, and sub-program executor. With the reasoning guidance G_t^*, the operator selector and operand selector generate an operator op_t and several operands od_t^i to form a sub-program. Then, the sub-program executor performs the numerical computation to obtain the intermediate result V_t (*i.e.*, process variable). Next, the intermediate result acts as the feedback signal by comparing with the ground truth answer (GT) to obtain the discrete rewards, and is also stored in the dynamic operand dictionary for subsequent sub-program generation.

When solving a geometry problem, humans usually read the text problem gradually and meanwhile align the information with the relevant geometry primitives in the accompanying diagram, which facilitates a better understanding of semantic details from local to global and multimodal perspectives. Inspired by this, we propose a Progressive Fusion Module (PFM) to model the relationships between the clause, the words in the clause, and the geometry diagram.

Text problem representation. The text problem is split into dense clauses by commas and periods for simplification, which makes it easier to read and understand the complex content for humans by dividing the text into smaller, more manageable chunks. These dense clauses can be denoted as $C = \{C_1, C_2, \ldots, C_K\}$ (K is the number of clauses), which provide a set of numerical and semantic information about the geometry primitives in the geometry diagram. The words in each clause C_k are represented with a text encoder, *e.g.*, BiGRU [33], RoBERTa [34], to obtain the representation $h_{k,n}$ for each word. Then, the context-enriched representation $\widetilde{C}_k$ for each clause is obtained by averaging all the word representations. The whole text problem with N_T words is finally represented as $H_T \in \mathbb{R}^{N_T \times d}$. Generally, the geometry problem starts with some contextual information about the text diagram and poses a question at last as a reasoning goal. Under the question guidance, the question-relevant information from the contextual clauses is captured to merge it into the question representation as follows:

$$s_k^q = \mathrm{softmax}(\mathbf{W}_{qc}[\widetilde{C}_k, q] + b_{qc}), \tag{1}$$

$$q^c = \sum_{k=1}^{K} s_k^q \widetilde{C}_k, \tag{2}$$

$$\widetilde{T} = \mathrm{ReLU}(\mathbf{W}_q[q^c, q] + b_q), \tag{3}$$

where $[\cdot,\cdot]$ denotes concatenation operation, $\mathbf{W}_{qc}$, $\mathbf{W}_q$, b_{qc} and b_q are learnable weight matrices and biases, ReLU denotes the activation function, and s_k^q denotes the relation weight between the kth clause and the question. Here, we take the representation of the final clause $\widetilde{C}_K$ in the text problem as the initial question representation, denoted as $q = \widetilde{C}_K$. Combining q and q^c, we can obtain the context-aware question representation $\widetilde{T} \in \mathbb{R}^d$ for the text problem.

Progressive fusion module. To enhance the diagram semantics, it is necessary to exploit the complementary information from the detailed text problem. The PFM grounds the geometry primitives in the diagram from both word and clause levels, which can enrich the diagram representation by explicitly performing the progressive feature fusion with the text semantics. Specifically, we adopt the diagram encoder from NGS [2], a ResNet-101 [35] pretrained by two auxiliary tasks. The extracted diagram representation is formalized as $H_D \in \mathbb{R}^{N_D \times d}$. We first model the relevance between the word-level representation in a clause and the diagram representation (*i.e.*, Word-Diagram Attention) as follows:

$$S_{d,w}(H_D, h_{k,n}) = \mathbf{W}_w^{n\top}\mathrm{ReLU}(\mathbf{W_1}[H_D, h_{k,n}] + b_1), \tag{4}$$

$$\alpha_{k,n}^w = \frac{\exp(S_{d,w}(H_D, h_{k,n}))}{\sum_{i=1}^{N_w} \exp(S_{d,w}(H_D, h_{k,i}))}, \tag{5}$$

where $h_{k,n}$ denotes the nth word representation in the kth clause, $\mathbf{W}_w$, $\mathbf{W}_1$, and b_1 are learnable weight matrices and biases, and $\top$ denotes the transpose operation.

Then, we model the relevance between each clause-level representation and the diagram representation H_D (*i.e.*, Clause-Diagram Attention) as follows:

$$S_{d,c}(H_D, \widetilde{C}_k) = \mathbf{W}_c^{\top}\mathrm{ReLU}(\mathbf{W_2}[H_D, \widetilde{C}_k] + b_2), \tag{6}$$

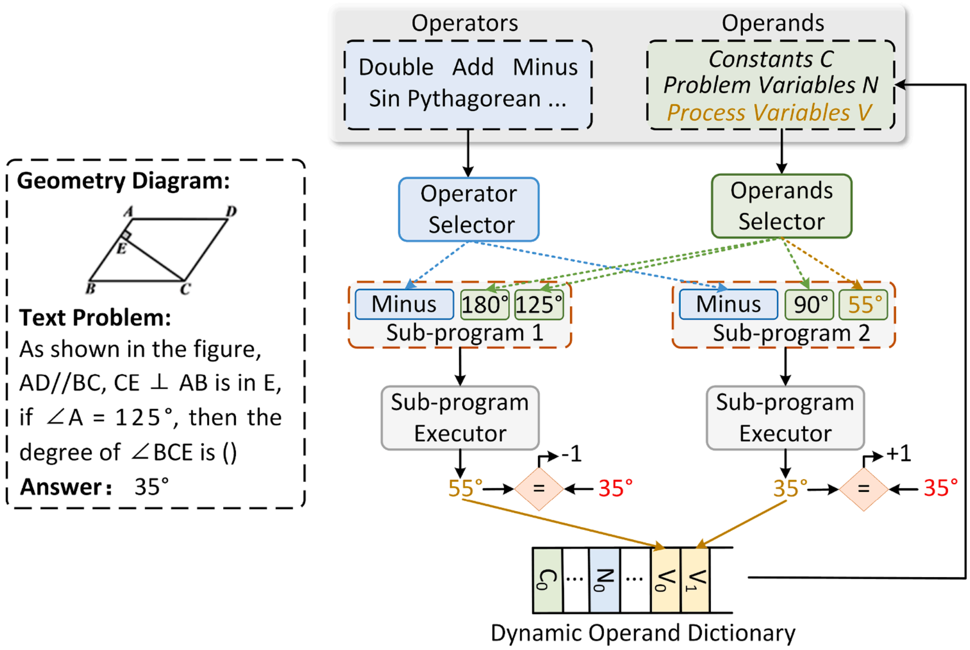

Fig. 3. One detailed example to demonstrate how the neural-symbolic reasoning decoder performs step-by-step reasoning.

$$\alpha_k^c = \frac{\mathbf{exp}(S_{d,c}(H_D, \widetilde{C}_k))}{\sum_{i=1}^{K} \mathbf{exp}(S_{d,c}(H_D, \widetilde{C}_i))}, \tag{7}$$

where $\mathbf{W}_c$, $\mathbf{W}_2$, and b_2 are learnable weight matrices and biases.

With the word-diagram attention $\alpha_{k,n}^w$ and the clause-diagram attention α_k^c at hand, the semantic-enhanced diagram representation $\widetilde{H}_D$ is finally generated as follows:

$$H_D^s = \sum_k \alpha_k^c (\widetilde{C}_k + \sum_n \alpha_{k,n}^w \cdot h_{k,n}), \tag{8}$$

$$\widetilde{H}_D = \mathrm{ReLU}(\mathbf{W}_d [H_D, H_D^s] + b_d), \tag{9}$$

where $\widetilde{H}_D$ incorporates the word-level and clause-level semantics from the text problem and also maintains the original visual semantics. Then, we concatenate the text representation H_T and the semantic-enhanced diagram representation $\widetilde{H}_D$ to obtain a comprehensive multimodal context representation $H^M \in \mathbb{R}^{(N_T+N_D)\times d}$ for guiding the subsequent solution program decoding.

Besides, we design a gate mechanism to obtain the compact multimodal joint representation S. It is adaptively selected from the context-aware question representation $\widetilde{T}$ and attention-based flatten diagram representation $\widetilde{H}_D^f$ via the gate mechanism, which assigns a probability to each of them as computed below:

$$\widetilde{H}_D^f = \mathrm{MLP}(\widetilde{H}_D), \tag{10}$$

$$gate_s = \sigma(\mathbf{W}_f [\widetilde{H}_D^f, \widetilde{T}] + b_f), \tag{11}$$

$$S = \mathbf{W}_s (gate_s \circ [\widetilde{H}_D^f, \widetilde{T}] + b_s, \tag{12}$$

where a two-layer multi-layer perceptron (MLP) is adopted to flatten the diagram representation $\widetilde{H}_D$ to $\widetilde{H}_D^f$, σ denotes the sigmoid activation function, and $\circ$ denotes the element-wise product. The compact multimodal joint representation S is later used as the initial hidden state for the decoder in the first decoding step. The comprehensive multimodal context representation H^M is fed to the decoder to guide the solution program generation step by step.

3.3. Neural-symbolic reasoning decoder

Instead of regarding the GPS task as a simple classification task similar to the general visual question answering task, the existing works [2–4,26] formulated it as a task of generating the solution program sequence with the help of heuristic program annotation. In such a way, the generated program can provide interpretability by reflecting the reasoning process. However, annotating such solution programs is a labor-intensive and time-consuming task [2]. In addition, students intuitively use diverse reasoning processes to solve the geometry problem in practice, hence fitting the only solution program limits the model's flexibility to explore other reasonable solutions. Therefore, we design a new neural-symbolic reasoning decoder to generate a solution program toward the final answer in the absence of other additional supervision. Different from the solution program decoder in [2–4,26] that generates the complete programs sequentially from left to right, our neural-symbolic reasoning decoder performs geometry reasoning step by step, where each step involves neural sub-program generation and symbolic sub-program execution for numerical computation as demonstrated in Fig. 3. By decoupling and performing the geometry reasoning process step by step, the intermediate result can act as the feedback signal via matching with the ground truth answer, further promoting the sub-program generation toward the final answer.

Specifically, we construct an operator dictionary with operators such as arithmetic and theorem operators (*e.g.,* multiply, add, Pythagorean minus). We also maintain a dynamic operand dictionary for operands that includes the constants C (*e.g.,* 90°, 180°, π), problem variables N (*i.e.,* the numbers in the text problem), and process variables V (*i.e.,* the intermediate executed results). At each step, under the reasoning guidance, the decoder generates a sub-program with an operator and several operands, and then computes a result for the sub-program. Since a complete solution program may contain several sub-programs, the dynamic operand dictionary enables us to use the executed results from the previous sub-programs in the subsequent ones. The neural-symbolic reasoning decoder mainly consists of Reasoning Guidance, Operator Selector, Operand Selector, and Sub-program Executor as shown in Fig. 2. The details are introduced as follows:

Reasoning guidance. For tth sub-program generation step, the reasoning guidance G_t^* for the operator or operands is calculated based on the attention between the comprehensive multimodal context representation H^M and previously generated decoding hidden state h_{t-1}^*:

$$S(h_{t-1}^*, h_i^M) = h_{t-1}^{*\top} \mathbf{W}_h \cdot \mathbf{W}_m h_i^M, \tag{13}$$

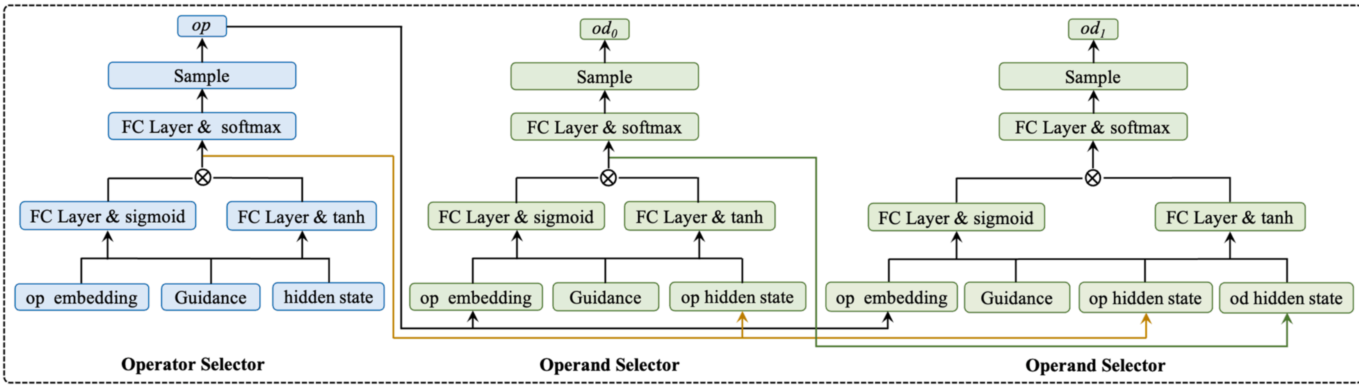

Fig. 4. The details of the neural network structure for the sub-program generation in the neural-symbolic reasoning decoder.

$$\alpha_i = \frac{\exp(S(h^*_{t-1}, h^M_i))}{\sum_j \exp(S(h^*_{t-1}, h^M_j))}, \tag{14}$$

$$G^*_t = \sum_i \alpha_i h^M_i, \tag{15}$$

where $\mathbf{W}_h$ and $\mathbf{W}_m$ are trainable parameters, $h^M_i \in H^M$, and h^*_{t-1} denotes the decoding hidden state for the operator or operands such as h^{op}_{t-1}.

Operator selector. The operator selector aims to decide which operator from the operator dictionary is selected. Here, we represent the operator dictionary as a learned embedding matrix $\mathbf{E}_{op} \in \mathbb{R}^{d_r \times N_{op}}$. Specifically, for the tth sub-program decoding step, the operator sampled in the last step op_{t-1}, the reasoning guidance G^{op}_t and the hidden state h^{op}_{t-1} are fed into the operator selector. Then, the probability of the tth operator is computed as follows:

$$g^{op}_t = \sigma\left(\mathbf{W}^1_{op}\left[\mathbf{W}^2_{op}[\mathbf{E}_{op}(op_{t-1}), G^{op}_t], h^{op}_{t-1}\right]\right), \tag{16}$$

$$h^{op}_t = g^{op}_t \cdot \tanh\left(\mathbf{W}^3_{op}\left[\mathbf{W}^4_{op}[\mathbf{E}_{op}(op_{t-1}), G^{op}_t], h^{op}_{t-1}\right]\right), \tag{17}$$

$$P^{op}_t = \mathrm{softmax}(\mathbf{W}^5_{op} h^{op}_t), \tag{18}$$

where $\mathbf{E}_{op}(op_{t-1})$ denotes the embedding of op_{t-1}, σ denotes the sigmoid activation function, and $\mathbf{W}^1_{op}$, $\mathbf{W}^2_{op}$, $\mathbf{W}^3_{op}$, $\mathbf{W}^4_{op}$, $\mathbf{W}^5_{op}$ are learnable matrices. The operator op_t is sampled from the probability distribution P^{op}_t.

Operand selector. Based on the decoded operator op_t, the operand selector utilizes the current operator embedding $\mathbf{E}_{op}(op_t)$, the reasoning guidance $G^{od}_{t,0}$ and the operator hidden state h^{op}_t to obtain a probability distribution $P^{od}_{t,0}$ of the first operand over the operand dictionary as follows:

$$g^{od}_t = \sigma\left(\mathbf{W}^1_{od}\left[\mathbf{W}^2_{od}[\mathbf{E}_{op}(op_t), G^{od}_{t,0}], h^{op}_t\right]\right), \tag{19}$$

$$h^{od}_{t,0} = g^{od}_t \cdot \tanh\left(\mathbf{W}^3_{od}\left[\mathbf{W}^4_{od}[\mathbf{E}_{op}(op_t), G^{od}_{t,0}], h^{op}_t\right]\right), \tag{20}$$

$$P^{od}_{t,0} = \mathrm{softmax}(\mathbf{W}^5_{od} h^{od}_{t,0}), \tag{21}$$

where $\mathbf{W}^1_{od}$, $\mathbf{W}^2_{od}$, $\mathbf{W}^3_{od}$, $\mathbf{W}^4_{od}$, $\mathbf{W}^5_{od}$ are learnable matrices. The operand od^0_t is sampled from the probability distribution $P^{od}_{t,0}$. When decoding the subsequent operands od^i_t, besides the operator embedding and the hidden state from the operator selector, the previous operands' hidden states are also used as inputs via a further concatenation operation for decoding. The details of the neural network structure for the operator selector and operand selector are demonstrated in Fig. 4.

Sub-program executor. After generating a sub-program, the executor performs symbolic numerical computation to obtain an intermediate result as a process variable. Specifically, the executor employs several symbolic modules to implement the corresponding operations on the operands to output a result. The computed intermediate result is appended to the dynamic operand dictionary (DODict), providing subsequent sub-program decoding steps with the opportunity to access it. Our DODict also limits the operand candidates of the problem variables N to the numbers that appear in the current problem, which can help narrow down the search space of the decoder and assist model training.

3.4. Objective optimization

Only the final answer is provided for training, but the explicit solution process for a problem is as necessary as the final answer. The model is expected to perform explicit step-by-step reasoning to generate a solution. Here, incorporating explicit reasoning as discrete operations in neural networks faces the non-differentiable problem, which poses optimization challenges. Therefore, we set our reasoning decoder in a reinforcement learning [36] framework with the answer as the supervision signal. The model is optimized with discrete binary feedback $R_t \in \{-1, +1\}$ from the exact match between the ground truth (GT) answer and the predicted numerical answer. Specifically, the sub-program with an operator and several operands denotes an action $\mathcal{A}$. As demonstrated in Fig. 3, after generating a sub-program, the sub-program is executed to obtain a result. Then, the predicted result from the sub-program is compared to the GT answer "35°". If the predicted result does not match the GT answer, a reward of -1 is given, the generated result is added to the dynamic operand dictionary as a process variable, and sub-program decoding continues. If the predicted result matches the GT answer, a reward of $+1$ is assigned, and the generation of the solution program is terminated. In this way, the GT answers serve as the supervision signal to provide training feedback by comparing with the sub-program executed results. By comparing the intermediate result to a known result, we can assess its accuracy and make necessary adjustments for future decoding steps. We use the REINFORCE objective to maximize the likelihood of actions that lead to the correct answer. In addition, we employ beam search to explore more candidate solution programs in the search space, alleviating the cold start problems and facilitating model training. In each sub-program decoding step, instead of sampling only one sub-program, we sample N_B sub-programs from the multinomial distribution based on the combinations of the operator and operands constructed from the output distribution of the operator and operand selector. The action sampling probability $P_\theta(\mathcal{A}_t)$ at the tth step is a product of the probability of the sampling operator P^{op}_t and the probability of the sampling operands $P^{od}_{t,i}$. The loss is defined as follows:

$$\mathcal{L}(\theta) = -\sum_t \mathbb{E}_{P_\theta(\mathcal{A}_t)} R_t, \tag{22}$$

where $\mathbb{E}$ denotes the expectation operation, and the goal is to maximize the sum of expected rewards R_t for choosing actions $\mathcal{A}_t$ by optimizing the parameter θ.

Table 1
Dataset statistics of four benchmark datasets including GeoQA, GeoQA + , Geometry3K and PGPS9K.

Datasets	GeoQA	GeoQA +		Geometry3K	PGPS9K
		Mix-train	Backtrans-train		
Train	3509	6027	12,054	8432	8021
Val	746	746	746	–	–
Test	755	755	755	589	1000
Total	5010	7528	13,555	9021	9021

4. Experiment

In this section, we first introduce the four datasets and the implementation details in Sections 4.1 and 4.2. Then, we present the comparison results with the state-of-the-arts in Section 4.3. Subsequently, we evaluate the effectiveness of NS-GPS based on several ablation studies in Section 4.4. Finally, we further make analyses for some case studies about success cases and failure cases in Section 4.5.

4.1. Datasets

We conduct experiments on four GPS datasets including GeoQA [2], GeoQA + [3], Geometry3K [5] and PGPS9K [28].

GeoQA and GeoQA + datasets. There are mainly three problem types including angle calculation, length calculation, and others that contain various types of problems such as area calculation. The statistics of the two datasets are shown in Table 1. The specific details are introduced as follows:

- The GeoQA dataset has a total of 5010 geometry problems, including train, val, and test set. The training set contains 3509 geometry problems of different problem types. The validation and test sets contain 746 and 755 problems, respectively.
- The GeoQA + dataset includes Mix-train, Backtrans-train, val, and test set. GeoQA + mainly expands the training set of GeoQA to form Mix-train and Backtrans-train. Specifically, the Mix-train of GeoQA + dataset contains 6027 geometry problems, which expands the GeoQA-train dataset with more difficult problems and richer problem types. Furthermore, the GeoQA + dataset is expanded to 12,054 geometry problems by performing a back-translated data augmentation method to obtain more diverse data. GeoQA and GeoQA + use the same validation and test sets. GeoQA + also constructs a new test set, which has the same size as the GeoQA-test set by extracting data from their newly annotated data. The answer accuracy is used as the evaluation metric.

Geometry3K and PGPS9K datasets. Compared with GeoQA and GeoQA + datasets, the two datasets all provide fine-grained diagram annotation in text form to convey rich information about diagrams. The Geometry3K dataset is available in two versions, provided by PGPSNet [28] and Inter-GPS [5], respectively. We adopt the version provided by PGPSNet, which shares similar annotation formats for diagrams and solution programs with the PGPS9K dataset. Specifically, the PGPS9K dataset [28] consists of 9021 geometry problems on 4000 geometry diagrams. Among them, 2891 problems about 1738 diagrams are selected from the Geometry3K dataset [5]. We adopt the same split ways for the PGPS9K dataset as [28] for fair comparison. The first way is to leave out the test set of Geometry3K containing 589 samples as the test set, and take the remaining 8432 samples as the training set. The second way is to partition the samples in a ratio of 8:1 based on the problem types, obtaining a training set with 8021 samples and a test set with 1000 samples. The two split results are denoted as Geometry3K and PGPS9K for method evaluation in the following experiments. The choice accuracy is adopted as the evaluation metric, defined as selecting the correct choice from four candidate options or choosing one randomly if the answer is not among them.

4.2. Implementation details

Our proposed NS-GPS is implemented based on the Pytorch platform. The diagram encoder adopts the pretrained ResNet101 [35], feeding with diagram images resized as 224×224. We apply the word2vec method to embed the words in the text problem. The dimension of the word embedding is set to 256. The text encoder adopts a two-layer Bi-GRU with an input embedding size of 256 and a hidden state size of 512. We also perform an ablation study with the pre-training model RoBERTa as the text encoder following [3]. The beam size N_B is set to 10. For the operator dictionary, there are 18 operators for the GeoQA dataset and 19 operators for the GeoQA + dataset. We choose Adam optimizer with weight decay $1e^{-5}$ to optimize our model. The initial learning rate is set to $1e^{-3}$, and the batch size is set to 8 to train a total of 30 epochs. We run our code with a single NVIDIA GeForce RTX 3090 GPU.

4.3. Experimental results

4.3.1. Performance comparison on GeoQA and GeoQA +

To illustrate the performance of our proposed NS-GPS on GeoQA and GeoQA + , we compare it with several baselines, which can be divided into two categories as follows:

- **With solution programs as supervision.** BERT2Prog [37], Seq2Prog [38], NGS [2], DPE-NGS [3] and SCA-GPS [26] convert the question answering task to a sequence generation task with the annotated programs acting as supervision. These methods achieve better performance and certain interpretability by introducing the solution programs for geometry problem solving. Specifically, BERT2Prog and Seq2Prog simply concatenate the text representation and diagram representation to obtain the multimodal representation. NGS has better diagram representation and multimodal reasoning ability by introducing two auxiliary tasks and a multimodal fusion method. DPE-NGS improves the text representation ability of NGS by combining RoBERTa and Bi-LSTM. SCA-GPS proposes two auxiliary tasks to enhance diagram representation and a Character-Diagram Feature Alignment module to obtain diagram-aware character representations by aligning them with corresponding geometry diagram features. DualGeoSolver [27] constructs a knowledge-inference system, which retrieves relevant geometric knowledge from an external knowledge base and incorporates it into the reasoning process to promote the program generation.
- **With answers as supervision.** FiLM [39], RN [40], and MCAN [20] have achieved good performance on the VQA task, demonstrating their strong multimodal reasoning ability. The geometry problem solving task can be regarded as a simple classification task, similar to the visual question answering task.

Table 2 reports the comparison results on the GeoQA and GeoQA + datasets. We follow the results of some baselines on the test set of the GeoQA dataset reported in [2]. We also reproduce the baselines to obtain the results on the new test set of the GeoQA + dataset. We mainly make the following four observations: (1) Compared to the methods with answers as supervision such as FiLM, RN, and MCAN, our NS-GPS achieves a noteworthy improvement on the test set of GeoQA, thereby highlighting the strength of performing geometry reasoning in the answering process. The performance comparison also exhibits that although the target of the geometry problem solving task is similar to that of the visual question answering task, it is inappropriate to regard it as a simple classification task due to its nature of geometry reasoning. Moreover, restricted only to the final answer, it lacks crucial interpretability for the intermediate problem solving process in GPS. (2) Compared to the methods with the annotated solution programs as supervision, our NS-GPS with the sole answer supervision achieves comparable performance with previous SOTA DPE-NGS, SCA-GPS and DualGeoSolver. Especially, NS-GPS obtains a 0.71 % performance improvement on the length type problems compared with DPE-NGS on

Table 2
Performance comparison on different types of questions for the test set of GeoQA and the new test set of GeoQA+. * denotes that the diagram encoder using the pre-trained ResNet-101 from NGS [2] is replaced by the pre-trained ViT from SCA-GPS [26] to provide the initial diagram feature.

Model	Param.	GeoQA				GeoQA+			
		Total	Angle	Length	Other	Total	Angle	Length	Other
With Solution Programs as Supervision									
BERT2Prog [37]	–	50.3	63.4	33.2	38.9	39.87	50.66	28.42	29.74
Seq2Prog [38]	–	52.6	63.6	39.2	37.0	40.53	50.93	34.97	25.64
NGS [2]	76.40M	60.7	72.0	47.0	44.4	49.14	53.85	45.90	43.07
DPE-NGS [3]	192.40M	62.7	74.9	47.7	50.0	51.52	54.64	49.18	47.69
SCA-GPS [26]	278.64M	64.1	74.9	50.1	55.5	52.45	55.17	49.18	49.74
DualGeoSolver [27]	280.69M	64.77	74.16	54.06	48.15	53.91	56.50	53.55	49.23
PFM-NGS	86.33M	63.18	73.92	49.82	50.00	52.32	55.44	51.37	47.18
PFM-NGS*	169.01M	64.37	75.36	50.53	51.85	53.51	56.23	53.01	48.72
With Answers as Supervision									
FiLM [39]	–	31.7	34.0	29.7	24.1	28.21	30.50	28.42	23.59
RN [40]	–	38.0	42.8	32.5	29.6	31.92	36.87	27.32	26.67
MCAN [20]	–	39.7	45.0	34.6	25.9	33.25	39.26	30.05	24.62
NS-GPS	75.76M	60.79	71.05	48.41	46.30	50.46	53.85	48.09	46.15
NS-GPS*	158.43M	61.72	72.25	48.76	48.15	51.26	54.11	49.73	47.18

the GeoQA-test set. This reveals the strength of NS-GPS, which achieves a balance between performance, annotation cost, and interpretability. Considering the high cost of exhaustive program annotations for geometry problems, which demands annotators with critical geometric knowledge, thus it is a promising and challenging direction to design geometry problem solvers with readily available answers as supervision. (3) In addition, our PFM is a plug-and-play module to help enhance multimodal semantic understanding. PFM-NGS achieves a better performance than DPE-NGS, which validates the superiority of our PFM design for promoting the multimodal reasoning ability of NGS. Moreover, our PFM-NGS and NS-GPS achieve performance improvement when using the pretrained ViT model from [26] as the diagram encoder. This demonstrates that the effectiveness of our model can be amplified with a stronger initial diagram representation. We analyze that a stronger visual representation backbone can better extract useful features from the diagram and reduce information loss, thereby providing higher-quality input for subsequent multimodal semantic interaction. Furthermore, the performance of PFM-NGS* is slightly lower than that of DualGeoSolver, which benefits from the incorporation of additional geometric knowledge in the reasoning process. Integrating such domain-specific knowledge into the model could be a promising direction for future performance improvement. (4) Compared to NGS, our PFM-NGS achieves significant performance gains with a comparable number of parameters (86.33M *vs*. 76.40M). Compared to DPE-NGS, our PFM-NGS achieves competitive performance with significantly fewer parameters (86.33M *vs*. 192.40M). Moreover, PFM-NGS with the pretrained ViT model as the diagram encoder (denoted as PFM-NGS*) exhibits superior performance with a comparable number of parameters to DPE-NGS (169.01M *vs*. 192.40M). NS-GPS* demonstrates highly competitive performance with a comparable number of parameters to DPE-NGS (158.43M *vs*. 192.40M). These experimental results highlight the effectiveness and superiority of our model design. Note that, in other experiments, we still adopt the parameter-efficient pretrained ResNet101 as our diagram encoder.

To test the generalization performance, we train our NS-GPS on the GeoQA-train set and also test on the new test set with the same size as GeoQA-test following [3]. Because the new test set is randomly extracted from the newly annotated difficult data in the GeoQA+ dataset, all solvers show a decreased performance compared with their performance on the GeoQA-test set as shown in Table 2. Nevertheless, our NS-GPS also achieves a competitive accuracy, which indicates the superior generalization ability of our model.

To sufficiently verify the superiority of NS-GPS, we use different training sets including GeoQA-train, Mix-train, and Backtrans-train, and set the GeoQA-test as the test set. As shown in Table 3, NS-GPS performs consistently well in all three settings. From rows 3, 6, and 9, we can observe that the accuracy improves when training our NS-GPS with Mix-train and Backtrans-train, which indicates that richer and more training samples can help promote the performance of our model. It is worth noting that these geometry problems, as well as their final answers, can be more easily mined from the internet compared to solution programs, which can facilitate the research of geometry problem solving.

Table 3
Performance comparison on the GeoQA-test set using the different training sets of GeoQA and GeoQA+ datasets.

Train Sets	Model	Total	Angle	Length	Other
GeoQA-train	NGS [2]	60.66	72.01	47.00	44.44
	DPE-NGS [3]	62.65	74.88	47.70	50.00
	NS-GPS	60.79	71.05	48.41	46.30
Mix-train	NGS [2]	61.19	72.25	47.70	46.30
	DPE-NGS [3]	65.96	75.60	54.42	51.85
	NS-GPS	62.12	73.21	48.40	48.15
Backtrans-train	NGS [2]	63.31	72.97	53.00	42.60
	DPE-NGS [3]	66.09	76.08	55.12	46.30
	NS-GPS	63.18	72.97	51.94	46.30

4.3.2. Performance comparison on Geometry3K and PGPS9K

To validate the generalization ability and effectiveness of our NS-GPS, we have also conducted experiments on the two datasets Geometry3K and PGPS9K, both provided by PGPSNet [28]. For a fair comparison, PGPSNet re-implements the baselines of NGS and Geoformer by preserving their pre-trained diagram encoder, taking both semantic clauses and textual problems as the text input, and applying the same data augmentation techniques. Following this setting, our model is also evaluated on Geometry3K and PGPS9K datasets. The main results are summarized in Table 4. We can make the following two observations:

- Compared with the similar neural solvers NGS and Geoformer, our PFM-NGS achieves superior performance improvement. It also outperforms the best baseline PGPSNet†. This demonstrates the effectiveness of PFM in promoting multi-modal semantic understanding.
- Compared to these methods with the annotated solution programs as supervision, our NS-GPS with the sole answer supervision also achieves comparable performance as NGS, Geoformer, and PGPSNet†. The experimental results on all four different datasets of GeoQA, GeoQA+, Geometry3K, and PGPS9K demonstrate the good generalization ability of our method.

Table 4
Performance comparison on Geometry3K and PGPS9K datasets. † denotes removing the pre-trained LM. The results of the baselines are excerpted from [28].

Methods	Geometry3K	PGPS9K
Baseline (Neural Solver) [5]	35.9	–
Inter-GPS (Predict) [5]	56.9	–
NGS [2]	58.8	46.1
Geoformer [4]	59.3	47.3
PGPSNet† [28]	61.7	–
PFM-NGS (ours)	63.2	53.5
NS-GPS (ours)	60.3	49.2

Table 5
Performance comparison on Geometry3K and PGPS9K datasets. v1 denotes integrating the pretrained LM from [28] as the text encoder.

Methods	Geometry3K	PGPS9K
Inter-GPS (Diagram GT) [5]	71.7	68.0
PGPSNet [28]	77.9	70.4
PFM-NGS (v1)	78.4	69.6
NS-GPS (v1)	74.9	67.5

Additionally, we incorporate a pre-trained LM from PGPSNet as our text encoder, referring to them as PFM-NGS (v1) and NS-GPS (v1). The experimental results are shown in Table 5. We can observe that both PFM-NGS (v1) and NS-GPS (v1) achieve significant performance improvements on both Geometry3K and PGPS9K datasets. We attribute these improvements primarily to the large-scale domain-specific pre-training, especially the semantic pre-training strategy, which enhances text representation capabilities and further facilitates the multimodal semantic understanding in the PFM module. In this setting, both PFM-NGS (v1) and NS-GPS (v1) can achieve comparable performance to the two baselines as shown in Table 5, which demonstrates the effectiveness and good generalization ability of our model. Moreover, these result analyses also inspire us that it is a promising research direction to pre-train a domain-specific language model by constructing a large-scale geometry-relevant corpus and developing some pre-training strategies in the future.

4.4. Ablation study

To study the impact of different components in NS-GPS, we conduct a series of thorough ablation studies on the GeoQA dataset as shown in Table 6.

Table 6
Ablation study of NS-GPS on GeoQA dataset.

Model	Total	Angle	Length	Other
NS-GPS	60.79	71.05	48.41	46.30
Importance of Components ↓				
w/o PFM	$59.21_{\downarrow 1.58}$	69.14	47.35	44.44
w/o Gate	$59.74_{\downarrow 1.05}$	70.33	46.64	46.30
w/o BiGRU	$59.87_{\downarrow 0.92}$	70.33	47.35	44.44
w/o Reasoning Guidance	$58.81_{\downarrow 1.98}$	68.90	47.00	42.59
Importance of Annotated Programs ↑				
w/ Part Programs	$62.52_{\uparrow 1.73}$	72.97	49.82	48.15
w/ All Programs	$64.50_{\uparrow 3.71}$	75.60	50.53	51.85
Importance of Decoder ↓				
w/o DODict	$63.44_{\downarrow 1.06}$	74.40	50.18	48.15
w/o NS Decoder	$63.18_{\downarrow 1.32}$	73.92	49.82	50.00

Importance of components. The variant w/o PFM directly applies the word-level representation to interact with the diagram following the encoder of NGS and DPE-NGS. We can observe that our full model shows about 1.58 % higher performance than the model without PFM, which indicates that the progressive multimodal fusion method helps enhance the multimodal semantic understanding. The variant w/o Gate removes the gate mechanism when fusing the text representation and the enhanced diagram representation. The performance drop reveals that selective visual-text fusion can help preserve the desired information and avoid redundant or noisy information, promoting program decoding. The variant w/o BiGRU replaces the BiGRU with the pre-training model RoBERTa as the text encoder, the performance shows a slight decrease. We analyze that the BiGRU has better representation ability for short clauses than RoBERTa, consistent with the analysis in [3]. The variant w/o Reasoning Guidance removes the dynamic reasoning guidance for each decoding step, and the decreased performance indicates that the reasoning guidance can provide relevant context information in the decoding process to help select the relative operators and operands.

Importance of annotated programs. In practical scenarios, ground-truth programs are generally unavailable for all problems. The performance improvement demonstrates that observing only a small part of all possible programs (30 %) can endow the model with a better ability to generate the reasoning solution programs compared with the setting of no ground-truth programs. In addition, we use ground-truth solution programs for all problems provided by the GeoQA dataset as strong supervision to train the neural-symbolic reasoning decoder. The results show that our NS-GPS with strong supervision can improve answer accuracy significantly.

Importance of decoder. When only the answers are provided for training, the components within the decoder are cohesive as a single unit and cannot be decoupled to conduct ablation studies. To demonstrate the effectiveness of each designed component in the decoder, we experiment based on the setting of NS-GPS w/ all programs displayed in a gray color row, and the last two rows display the ablated results. When replacing the dynamic operand dictionary (DODict) with a fixed dictionary about all vocabulary in the dataset, there is a drop of 1.06 % in performance compared with the full model NS-GPS w/ all programs. We analyze this because it does not limit the selection of reasoning process variables and also does not limit the search space of operands, leading to the difficulty of model training. When replacing our neural-symbolic reasoning decoder (NS Decoder) with an LSTM decoder to generate the solution program based on a seq2seq framework. The variant exhibits 1.32 % performance degradation, highlighting the advantage of separating the generation for operators and operands and performing the step-wise sub-program generation and execution.

4.5. Case study

We conduct case studies to analyze the strengths and weaknesses of our NS-GPS, as shown in Fig. 5. NS-GPS performs step-wise geometry reasoning via sub-program generation and execution. The multiple sub-programs can form a complete solution program. For case (a), our proposed NS-GPS can generate a correct solution program as the same as the annotated solution program, which can certainly be executed to obtain the correct answer. For case (b), NS-GPS can generate another reasonable solution program that can also be executed to obtain the ground-truth answer. It is common for a geometry problem to be solved via multiple solutions. However, existing methods such as NGS optimize the solution program generation in the training process by providing only one program, which limits the model's ability to explore diverse solutions. For case (c), NS-GPS generates a wrong solution program despite arriving at the correct answer. For case (d), it is necessary to take advantage of the property of similar triangles, that is, the ratio of the area of two similar triangles is equal to the square of the ratio of any pair of the corresponding sides. With the geometry context at hand, NS-GPS

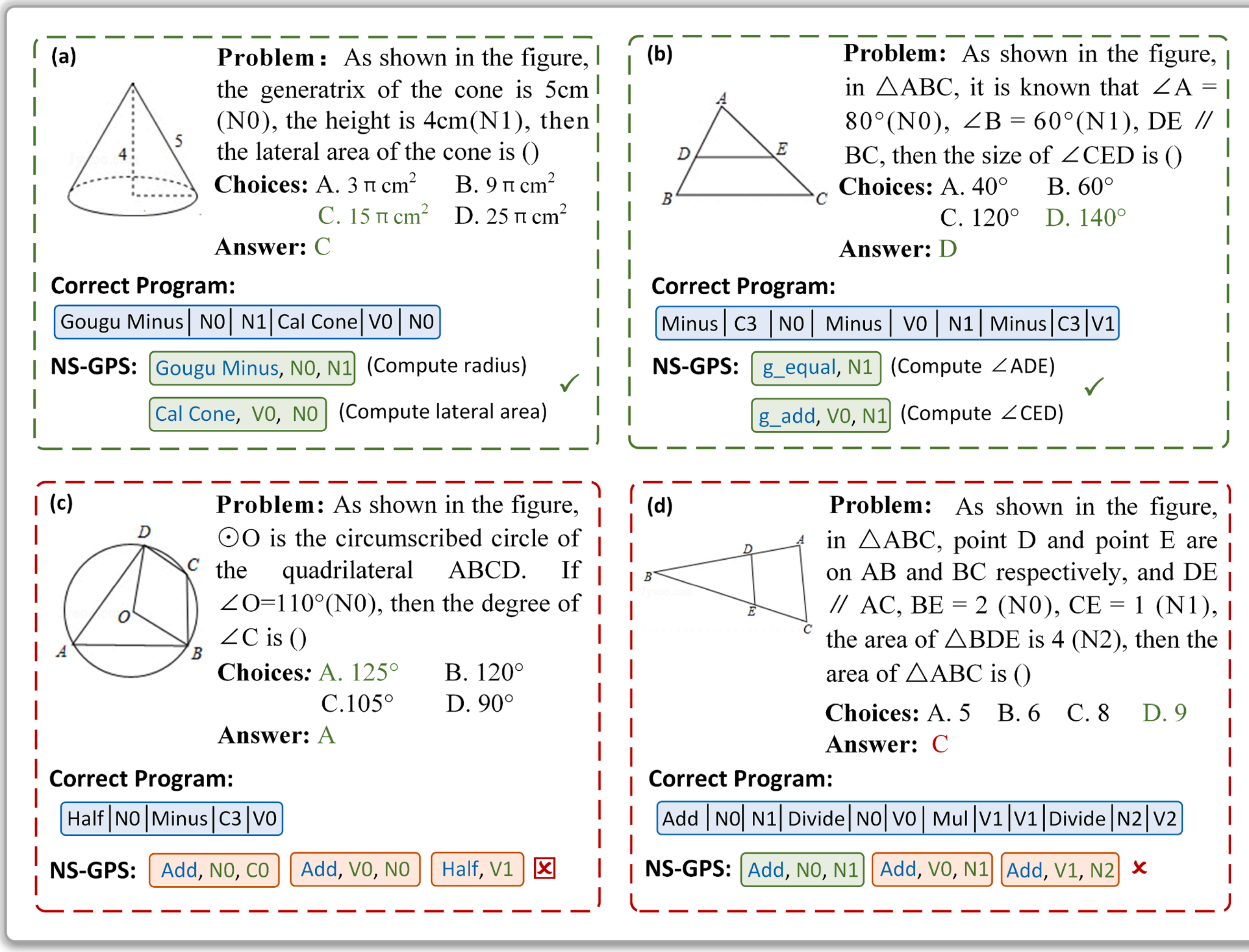

Fig. 5. Typical cases including success cases and failure cases. Case (a) is a correct case with the correct solution program, the same as the annotated program. Case (b) is a correct case with another reasonable solution program to reach the correct answer. Case (c) is a spurious case with a wrong solution program to reach the right answer. Case (d) is an error case with a wrong solution program to reach a wrong answer.

still struggles to decide to apply and correctly apply this high-level geometry theorem with complex operations, leading to an unreasonable solution and a wrong answer. Therefore, there is still great potential to improve the complex multimodal geometry reasoning ability and introduce sufficient theorem knowledge, which sheds light on promising directions for our future research.

This case-based qualitative analysis intuitively demonstrates the interpretability of our model, which provides the intermediate solution programs. Additionally, we conduct an additional quantitative analysis focusing on the solution program sequences generated by NS-GPS to further evaluate the interpretability. Specifically, we randomly sampled 50 geometry problems from the subset of correctly solved instances in the test set and manually evaluated their corresponding solution programs using the following two criteria: (1) Logical coherence and rigor. Whether the sequence of symbolic programs follows a valid geometric reasoning flow, with each step logically connected to the previous ones and grounded in correct geometric principles. (2) Alignment with human reasoning patterns. Whether the overall problem solution reflects common and rational strategies used by humans. The result shows that 94 % of the program sequences satisfied both criteria, demonstrating the method's robustness in producing logically sound and human-aligned solution programs.

5. Conclusion

In this work, we propose a novel Neural-Symbolic Geometry Problem Solver (NS-GPS) to solve geometry problems through step-by-step reasoning. It is trained end-to-end in a reinforcement learning framework with discrete rewards from matching the ground truth and predicted answer. For the multimodal representation encoder, a progressive fusion module is designed to model the interactions between the geometry diagram and the word-level, clause-level semantics information in the text problem, further enhancing the multimodal semantics understanding of the geometry context. After that, a neural-symbolic reasoning decoder is proposed to perform step-wise geometry reasoning consisting of neural sub-program generation and symbolic numerical computation. We utilize the intermediate results of each sub-program as training feedback to promote the sub-program generation toward the final answer. Moreover, the design of the dynamic operand dictionary helps limit the search space for each decoding step, promoting model training. Most importantly, NS-GPS can generate intermediate solution programs as the explicit reasoning process without additional supervision, avoiding labor-intensive solution program annotations. Extensive experimental results and ablation studies verify the effectiveness and superiority of our proposed NS-GPS.

Limitations. While our method demonstrates promising results, there are still several limitations that warrant further exploration: (1) Lack of detailed theorem knowledge integration. Geometry problem solving is inherently a theorem-driven task. However, the current method does not explicitly integrate detailed theorem knowledge, which may limit its accuracy in answering questions. For example, the Inscribed Angle Theorem states that an inscribed angle in a circle is always half of the corresponding central angle subtending the same arc. Investigating how to incorporate detailed theorem knowledge in the answer reasoning process remains a crucial challenge. (2) Limited readability of solution programs. While our model is capable of generating structured solution programs with interpretability, there is still room for improvement in terms of readability. Compared to structured solution formats, natural language explanations are generally more readable and accessible to humans. However, ensuring that these explanations remain mathematically rigorous and verifiable is challenging. Geometry solutions require a high level of standardization and strict adherence to formal mathematical language. Future research should explore methods to enhance the generation of high-quality solution processes that balance interpretability, readability, and formal verifiability.

CRediT authorship contribution statement

Yaxian Wang: Validation, Investigation, Writing – original draft, Writing – review & editing, Visualization, Methodology, Conceptualization; **Bifan Wei:** Methodology, Funding acquisition, Writing – review & editing, Investigation, Supervision; **Yinghong Ma:** Validation, Writing – review & editing, Methodology, Investigation; **Xudong Jiang:** Conceptualization, Writing – review & editing; **Henghui Ding:** Writing – review & editing, Conceptualization; **Zhongmin Cai:** Writing – review & editing, Supervision; **Jun Liu:** Writing – review & editing, Funding acquisition, Supervision, Conceptualization, Methodology.

Data availability

Data will be made available on request.

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.

Acknowledgments

This work was supported by National Key Research and Development Program of China (2022YFC3303600), National Natural Science Foundation of China (62137002, 62293550, 62293553, 62293554, 62450005, 62437002, 62477036, 62477037, 62176209, 62192781, and 62306229), "LENOVO-XJTU" Intelligent Industry Joint Laboratory Project, Natural Science Basic Research Program of Shaanxi (2023-JC-YB-593), the Youth Innovation Team of Shaanxi Universities "Multimodal Data Mining and Fusion", the Shaanxi Provincial Social Science Foundation Project (2024P041).

References

[1] M. Seo, H. Hajishirzi, A. Farhadi, O. Etzioni, C. Malcolm, Solving geometry problems: combining text and diagram interpretation, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2015, pp. 1466–1476.
[2] J. Chen, J. Tang, J. Qin, X. Liang, L. Liu, E. Xing, L. Lin, GeoQA: a geometric question answering benchmark towards multimodal numerical reasoning, in: Findings of the Association for Computational Linguistics: ACL-IJCNLP, 2021, pp. 513–523.
[3] J. Cao, J. Xiao, An augmented benchmark dataset for geometric question answering through dual parallel text encoding, in: Proceedings of the International Conference on Computational Linguistics, 2022, pp. 1511–1520.
[4] J. Chen, T. Li, J. Qin, P. Lu, L. Lin, C. Chen, X. Liang, UniGeo: unifying geometry logical reasoning via reformulating mathematical expression, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2022, pp. 3313–3323.
[5] P. Lu, R. Gong, S. Jiang, L. Qiu, S. Huang, X. Liang, S.-c. Zhu, Inter-GPS: interpretable geometry problem solving with formal language and symbolic reasoning, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing, 2021, pp. 6774–6786.
[6] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, Proc. Adv. Neural Inf. Process. Syst. 27 (2014), pp. 3104–3112.
[7] E.J. St B. T, In two minds: dual-process accounts of reasoning, Trends Cogn. Sci. 7 (10) (2003) 454–459.
[8] E.J. St B. T, Dual-processing accounts of reasoning, judgment, and social cognition, Annu. Rev. Psychol. 59 (2008) 255–278.
[9] A. Masry, D. Long, J.Q. Tan, S. Joty, E. Hoque, ChartQA: a benchmark for question answering about charts with visual and logical reasoning, in: Findings of the Association for Computational Linguistics: ACL 2022, 2022, pp. 2263–2279.
[10] Y. Wang, J. Liu, J. Ma, H. Zeng, L. Zhang, J. Li, Dynamic dual graph networks for textbook question answering, Pattern Recognit. 139 (2023) 109441.
[11] M. Zhang, F. Yin, Y. Hao, C. Liu, Plane geometry diagram parsing, in: Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, 2022, pp. 1636–1643.
[12] S. Peng, D. Fu, Y. Liang, L. Gao, Z. Tang, GeoDRL: a self-learning framework for geometry problem solving using reinforcement learning in deductive reasoning, in: Findings of the Association for Computational Linguistics: ACL 2023, 2023, pp. 13468–13480.
[13] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C.L. Zitnick, D. Parikh, VQA: visual question answering, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 2425–2433.
[14] J. Lu, J. Yang, D. Batra, D. Parikh, Hierarchical question-image co-attention for visual question answering, in: Proceedings of Advances in Neural Information Processing Systems, 2016, pp. 289–297.
[15] Q. Wang, J. Liu, W. Wu, Coordinating explicit and implicit knowledge for knowledge-based VQA, Pattern Recognit. 151 (2024) 110368.
[16] Y. Wang, B. Wei, J. Liu, L. Zhang, J. Wang, Q. Wang, DisAVR: disentangled adaptive visual reasoning network for diagram question answering, IEEE Trans. Image Process. 32 (2023), pp. 4812–4827.
[17] F. Xu, Q. Lin, J. Liu, L. Zhang, T. Zhao, Q. Chai, Y. Pan, Y. Huang, Q. Wang, MoCA: incorporating domain pretraining and cross attention for textbook question answering, Pattern Recognit. 140 (2023) 109588.
[18] K. Kafle, B. Price, S. Cohen, C. Kanan, DVQA: understanding data visualizations via question answering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5648–5656.
[19] L. Ma, Z. Lu, H. Li, Learning to answer questions from image using convolutional neural network, in: Proceedings of the AAAI Conference on Artificial Intelligence, 30, 2016.
[20] Z. Yu, J. Yu, Y. Cui, D. Tao, Q. Tian, Deep modular co-attention networks for visual question answering, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 6281–6290.
[21] R. Hu, J. Andreas, T. Darrell, K. Saenko, Explainable neural computation via stack neural module networks, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 53–69.
[22] J. Mao, C. Gan, P. Kohli, J.B. Tenenbaum, J. Wu, The neuro-symbolic concept learner: interpreting scenes, words, and sentences from natural supervision, in: Proceedings of the International Conference on Learning Representations, 2019.
[23] K. Yi, J. Wu, C. Gan, A. Torralba, P. Kohli, J. Tenenbaum, Neural-symbolic VQA: disentangling reasoning from vision and language understanding, in: Proceedings of Advances in Neural Information Processing Systems, 2018, pp. 1039–1050.
[24] J. Johnson, B. Hariharan, L. Van Der Maaten, L. Fei-Fei, C. Lawrence Zitnick, R. Girshick, CLEVR: a diagnostic dataset for compositional language and elementary visual reasoning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2901–2910.
[25] M. Sachan, K. Dubey, E. Xing, From textbooks to knowledge: a case study in harvesting axiomatic knowledge from textbooks to solve geometry problems, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2017, pp. 773–784.
[26] M. Ning, Q.-F. Wang, K. Huang, X. Huang, A symbolic characters aware model for solving geometry problems, in: Proceedings of the 31st ACM International Conference on Multimedia, 2023, pp. 7767–7775.
[27] T. Xiao, J. Liu, Z. Huang, J. Wu, J. Sha, S. Wang, E. Chen, Learning to solve geometry problems via simulating human dual-reasoning process, in: Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, 2024, pp. 6559–6568.
[28] M.-L. Zhang, F. Yin, C.-L. Liu, A multi-modal neural geometric solver with textual clauses parsed from diagram, in: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, 2023, pp. 3374–3382.
[29] Z.-Z. Li, M.-L. Zhang, F. Yin, C.-L. Liu, LANS: a layout-aware neural solver for plane geometry problem, in: Findings of the Association for Computational Linguistics ACL 2024, 2024, pp. 2596–2608.
[30] B. Ma, P. Jian, C. Pan, Y. Wang, W. Ma, A geometric neural solving method based on a diagram text information fusion analysis, Sci. Rep. 14 (1) (2024) 31906.
[31] OpenAI, GPT-4V(ision) system card, https://openai.com/research/ gpt-4v-system-card (2023).
[32] J. Bai, S. Bai, S. Yang, S. Wang, S. Tan, P. Wang, J. Lin, C. Zhou, J. Zhou, Qwen-VL: a versatile vision-language model for understanding, localization, text reading, and beyond, arXiv preprint arXiv:2308.12966 (2023).
[33] A. Graves, A. Graves, Long short-term memory, Supervised Sequence Labelling with Recurrent Neural Networks (2012) 37–45.

[34] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: a robustly optimized BERT pretraining approach, arXiv preprint arXiv:1907.11692 (2019).
[35] 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.
[36] R.J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, Reinforcement Learn. 8 (1992), pp. 5–32.
[37] J.D. M.-W.C. Kenton, L.K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), 2019, pp. 4171–4186.
[38] A. Amini, S. Gabriel, S. Lin, R. Koncel-Kedziorski, Y. Choi, H. Hajishirzi, MathQA: towards interpretable math word problem solving with operation-based formalisms, in: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), 2019, pp. 2357–2367.
[39] E. Perez, F. Strub, H. De Vries, V. Dumoulin, A. Courville, Film: visual reasoning with a general conditioning layer, in: Proceedings of the AAAI Conference on Artificial Intelligence, 32, 2018.
[40] A. Santoro, D. Raposo, D.G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, T. Lillicrap, A simple neural network module for relational reasoning, in: Proceedings of Advances in Neural Information Processing Systems, 2017, pp. 4967–4976.