A Multi-Modal Neural Geometric Solver with Textual Clauses
Parsed from Diagram

Ming-Liang Zhang1,2, Fei Yin1,2, Cheng-Lin Liu1,2

1MAIS, Institute of Automation of Chinese Academy of Sciences
2School of Artificial Intelligence, University of Chinese Academy of Sciences
zhangmingliang2018@ia.ac.cn, fyin@nlpr.ia.ac.cn, liucl@nlpr.ia.ac.cn

Abstract
Geometry problem solving (GPS) is a high-level mathematical reasoning requiring the capacities of multi-modal fusion and geometric knowledge application. Recently, neural solvers have shown great potential in GPS but still be short in diagram presentation and modal fusion. In this work, we convert diagrams into basic textual clauses to describe diagram features effectively, and propose a new neural solver called PGPSNet to fuse multi-modal information efficiently. Combining structural and semantic pre-training, data augmentation and self-limited decoding, PGPSNet is endowed with rich knowledge of geometry theorems and geometric representation, and therefore promotes geometric understanding and reasoning. In addition, to facilitate the research of GPS, we build a new large-scale and fine-annotated GPS dataset named PGPS9K, labeled with both fine-grained diagram annotation and interpretable solution program. Experiments on PGPS9K and an existing dataset Geometry3K validate the superiority of our method over the state-of-the-art neural solvers. Our code, dataset and appendix material are available at https://github.com/mingliangzhang2018/PGPS.

1 Introduction
Automatic geometry problem solving (GPS) is a longstanding and challenging AI task, and has attracted much attention in the CV and NLP community recently [Seo et al., 2015; Sachan et al., 2017; Lu et al., 2021; Chen et al., 2021; Zhang et al., 2022]. A geometry problem is formed with a textual problem and a geometry diagram, where the textual problem describes the geometry problem condition and sets the reasoning objective in natural language, and the geometry diagram carries rich structural and semantic information beyond the textual problem to aid problem solving. GPS requires mathematical and multi-modal reasoning capabilities combining the textual problem and geometry diagram. The multi-modal integration and geometric knowledge application are the keys of GPS.

Existing works of GPS can be classified into two categories: symbolic geometric solvers [Seo et al., 2015; Sachan et al., 2017; Lu et al., 2021] and neural geometric solvers [Chen et al., 2021; Cao and Xiao, 2022; Chen et al., 2022]. The symbolic solvers parse the diagram and textual problem into a unified formal language. Based on the geometric theorem knowledge, they usually perform symbolic reasoning by path search and condition matching to produce new conditional states until they find the search target. While the symbolic solvers have better interpretability compared with neural solvers, they are carefully designed with complex rules and hard to extend. Besides, some symbolic solvers may solve problems slowly with many redundant steps, and the search process also does not match humans’ solutions. The neural solvers proposed recently embed the diagram and textual problem jointly with the hybrid encoder and self-supervised auxiliary tasks and generate the solution program in sequential form. Although neural solvers have achieved impressive results with simple pipelines, recent works [Lu et al., 2021] and neural geometric solvers [Chen et al., 2021; Cao and Xiao, 2022; Chen et al., 2022].
al., 2021; Lu et al., 2022] show that there remains a large performance gap compared with symbolic solvers. One of major reasons is that neural solvers, adopting similar frameworks of general multi-modal reasoning tasks applied for natural images, cannot exploit the diagram efficiently. Because primitives in geometry diagram are slender and overlapped, and have complex spatial relationship, the feature map based [Anderson et al., 2018], region proposal based [Yu et al., 2019] or image patch based [Kim et al., 2021] frameworks cannot extract fine-grained features and even damage structural and semantic information.

Considering the under-representation of diagram and difficulty of cross-modal fusion in neural solver, we represent the geometry diagram by textual clauses including structural clauses and semantic clauses, as demonstrated in bottom of Figure 1. Compared with visual image, clauses have highly syntactic structures with less redundant information naturally. Textual clauses are better suited to describe fine-grained and multi-level information in geometry diagram. Different from the formal language consisting of complex multi-order logic forms in symbolic solver, our clauses only describe basic relationships, where structural clauses depict the connection relations among geometric primitives and the semantic clauses describe the semantic relations between non-geometric primitives and geometric primitives. We do not pursue higher level relations constructed by geometric rules since they do not conform to the goal of neural solver. The model is hoped to have the ability of matching geometric patterns and constructing high-level relations from basic relations.

To promote the research on GPS, we also build a new large-scale and fine-annotated GPS dataset named PGPS9K. PGPS9K contains 9,022 geometry problems paired with geometry diagrams. In contrast to existing datasets, PGPS9K is labeled with both diagram annotation and solution program. The diagram annotation employs the same primitive level annotation method as [Zhang et al., 2022; Hao et al., 2022], which could be translated to textual clauses simply and uniquely. Given the solution complexity of GPS, we design a new annotation form based on geometric theorems for solution program. The solution program provides the problem solving procedure wherein each step is an application of a theorem (axiom), as shown in top of Figure 1, rather than the specific solution steps involving fundamental arithmetic operations [Wang et al., 2017; Amini et al., 2019]. Our solution program carries rich geometric knowledge and has better interpretability, and as well reduces the burden of model learning.

Taking advantage of multi-modal information, we thus propose a new diagram-text fusion neural solver named PGPSNet as shown in Figure 1. In addition to the geometry diagram and textual problem, PGPSNet combines structural clauses and semantic clauses and generates the solution program to solve problem. To fuse different parts of text modality, we propose a structural and semantic pre-training strategy based on Masked Language Modeling (MLM) task [Devlin et al., 2019], for improving the model’s understanding of structural and semantic content by explicit modeling. To overcome the limitation of data size and pre-trained corpus, we design five data augmentation strategies based on diversity and equivalence of geometric representation. Besides, we construct a self-limited GRU decoder to shrink the representation space and search space of operands and speed up training and inference. Experiments on an existing dataset Geometry3K [Lu et al., 2021] and PGPS9K dataset demonstrate that our PGPSNet boosts the performance of GPS prominently, largely exceeds the performance of existing neural solvers, and also achieves comparable results as well-designed symbolic solvers.

The contributions of this work are summarized in four folds: (1) We use textual clauses to express the fine-grained structural and semantic information in geometry diagram efficiently. (2) We propose a new neural solver PGPSNet, fusing multi-modal information through structural and semantic pre-training, data augmentation, and self-limited decoding. (3) We construct a large-scale dataset PGPS9K labeled with both fine-grained diagram annotation and interpretable solution program. (4) Our PGPSNet outperforms existing neural solvers significantly.

2 Related Work
2.1 Multi-modal Reasoning

Multi-modal reasoning uses multi-modal datasets to perform reasoning tasks, e.g., visual question answering [Anderson et al., 2018; Kim et al., 2021], document visual question answering [Xu et al., 2020; Tito et al., 2021], table question answering [Zhu et al., 2021; Lu et al., 2023], where GPS is also a special multi-modal reasoning task. Because of difference of data modality and reasoning ability, it results in significant semantic gaps between different tasks. The core to multi-modal reasoning lies on how to unite modalities and incorporate domain knowledge.

2.2 Geometry Problem Solving

Generalized GPS contains geometry calculation [Seo et al., 2015; Tsai et al., 2021] and geometry proving [Chou et al., 1996; Gan et al., 2019], and can be divided into symbolic solvers [Seo et al., 2015; Lu et al., 2021] and neural solvers [Chen et al., 2021; Chen et al., 2022] in method. The symbolic solvers that have been studied for years possess their advantages and limitations as introduced in Introduction. With development of neural network, neural solvers have shown their potential in GPS, whereas they still cannot handle modal representation and fusion well. In context of the geometry calculation, our work unifies the geometry diagram and textual problem into the text modality to better construct structural and semantic relationships in geometry.

2.3 Pre-trained Language Model for Mathematical Reasoning

Language models [Lewis et al., 2020; Brown et al., 2020], pre-trained on large text corpus with self-supervised learning tasks such as MLM [Devlin et al., 2019] or CLM [Lewis et al., 2020], have demonstrated remarkable performance gains on wide range of NLP tasks such as text classification [Minaee et al., 2021] and question answering [Khashabi et al., 2021]. Inspired by them, pre-trained language model is also applied to mathematical reasoning task gradually, e.g., math
word problem solving [Liang et al., 2022; Zhang and Moshfeghi, 2022] and GPS [Lu et al., 2021; Chen et al., 2022]. However, lacking of large-scale mathematical corpus and targeted pre-training tasks, existing language models pre-trained on natural corpus seem to have limited effects on downstream reasoning tasks with only fine-tune tactic. In this work, we pre-train language model with MLM task combining textual reasoning tasks with only fine-tune tactic. In this work, we build a new large-scale only for rule-based symbolic solvers, or coarse-grained proposed, as presented in Table 1, they are either in small scale only for rule-based symbolic solvers, or coarse-grained.

### Table 1: Comparison with existing GPS datasets. Type, OP and PL represent problem type, operator number and program length, respectively.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#QA</th>
<th>Grade</th>
<th>#Type</th>
<th>Diagram Anno</th>
<th>Rationale</th>
<th>#Avg OP</th>
<th>#Avg PL</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEOS [Seo et al., 2015]</td>
<td>186</td>
<td>6-10</td>
<td>-</td>
<td>No</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>GEOS++ [Sachan et al., 2017]</td>
<td>1,406</td>
<td>6-10</td>
<td>-</td>
<td>No</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>GEOS-OS [Sachan and Xing, 2017]</td>
<td>2,235</td>
<td>6-10</td>
<td>-</td>
<td>Demonstration</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Geometry3K [Lu et al., 2021]</td>
<td>3,002</td>
<td>6-12</td>
<td>4</td>
<td>Yes</td>
<td>Logical form</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>GeoQA [Chen et al., 2021]</td>
<td>4,998</td>
<td>6-12</td>
<td>3</td>
<td>No</td>
<td>Program</td>
<td>1.98</td>
<td>5.35</td>
</tr>
<tr>
<td>GeoQA+ [Cao and Xiao, 2022]</td>
<td>7,528</td>
<td>6-12</td>
<td>3</td>
<td>No</td>
<td>Program</td>
<td>2.61</td>
<td>-</td>
</tr>
<tr>
<td>PGPS9K</td>
<td>9,022</td>
<td>6-12</td>
<td>30</td>
<td>Yes</td>
<td>Program</td>
<td>2.43</td>
<td>7.45</td>
</tr>
</tbody>
</table>

Figure 2: Example presentation of PGPS9K dataset.

3 PGPS9K Dataset

Although several datasets [Seo et al., 2015; Sachan et al., 2017; Lu et al., 2021; Chen et al., 2022] for GPS have been proposed, as presented in Table 1, they are either in small scale only for rule-based symbolic solvers, or coarse-grained annotated neglecting rich information in diagram. To facilitate the application of neural solver, we build a new large-scale GPS dataset called PGPS9K labeled both fine-grained diagram annotation and interpretable solution program. To the best of our knowledge, PGPS9K is the largest and the most complete annotation dataset for GPS up to now.

3.1 Collection and Description

PGPS9K is composed of 9,022 geometry problems paired with non-duplicate 4,000 geometry diagrams, where 2,891 problems paired with 1,738 diagrams are selected from Geometry3K dataset [Lu et al., 2021], the rest of problems are collected from five popular textbooks across grades 6-12 on mathematics curriculum websites. Our PGPS9K is divided into 30 problem types as exhibited in appendix A.1, covering almost all problem types of plane geometry problem in corresponding grades. As shown in Figure 2, PGPS9K dataset has five properties: (1) **Theorem-based**: Solving problems in PGPS9K need to apply geometric theorem knowledge to carry out algebraic calculation and get numerical results finally; (2) **Diagram-dependent**: Above 90% of problems must be solved using the diagrams because necessary conditions such as variable content and geometric structure are displayed via visual form instead of text; (3) **Abstract**: The diagram is integrated with basic geometric primitives (point, line, circle) and non-geometric primitives (text, symbol). No complex semantic scenarios are involved in textual problem except abstract geometric conditions; (4) **Fine-grained**: Problems with the same diagram vary in conditions or targets. Slight distinctions in textual problems usually lead to completely different solutions to problems; (5) **Condition-redundancy**: Lots of conditions in semantic clauses or textual problem are not needed in problem solving at hand. These five properties make PGPS9K focus on the challenges at geometric reasoning and alleviate the bias introduced by the text [Manjunatha et al., 2019; Patel et al., 2021]. Moreover, for convenience of experimental comparison, we split PGPS9K in two ways: The first is leaving out the test set of Geometry3K [Lu et al., 2021] as test set (589) and other disjoint samples as training set (8,433); The second is dividing samples of each problem type accord-

---

1[http://www.nlpr.ia.ac.cn/databases/CASIA-PGPS9K](http://www.nlpr.ia.ac.cn/databases/CASIA-PGPS9K)

2[https://www.mheducation.com/](https://www.mheducation.com/)
Figure 3: Annotation of solution program and its interpretability.
Fine-grained class tags and section tags promote the modeling of geometry problems and alleviate the imbalance of textual token. Then we mask 30% of text tokens with mask tokens following [Cho et al., 2021] but keep tags unchanged. The model is trained to recover the masked text in a unified text generation manner.

This pre-training method is extremely applicable to structural and semantic modeling of geometry under our modal representation. For instance, it can be reasoned that the mask token in the semantic clause "BD ⊥ EA on [M]" is "C" according to structural clauses "line B C D" and "line E C A", facilitating model learning the geometric knowledge of line intersection. However, in some cases, model may not infer the mask content exactly but geometric knowledge is filled in its token candidates. Taking instance of the textual problem "Find [M]D.", there is high probability that the mask token is "C" or "B" according to the structural clause "line B C D". In summary, pre-training makes model acquire advanced geometric cognition that is an elementary skill for GPS.

4.3 Encoder and Decoder

In our model, the CNN encoder only extracts coarse-grained global visual features of diagram such as geometric style, to determine possible operations quickly and accelerate model convergence. The GRU encoder integrates the diagram encoded as visual token and textual tokens enhanced by structural and semantic language model, and obtains the mixed encoding context $H^E = \{h^E_i\}_{i=1}^{n+1}$ as output.

Because of complexity and flexibility of solution process of geometry problem, solution program cannot convert into binary or general expression tree. Tree decoders widely used in math word problem [Xie and Sun, 2019; Tsai et al., 2021] are not applicable to GPS. We propose a self-limited GRU decoder to generate the sequential solution program in an autoregressive manner. The differences between self-limited decoder and the general attention-based decoder [Bahdanau et al., 2015] are two folds: (1) Self-limited decoder reduces token embedding space. The input embeddings of problem variables $N$ and augments $ARG$ presented in problem text are copied from encoder output, which also enriches decoder inputs with contextual semantic. Specifically, token embeddings are defined as:

$$e(y) = \begin{cases} \text{TokenEmb}(y), & y \in \{V_Y, V_C\}, \\ h^E_{loc(y,T)}, & y \in \{V_N, V_{ARG}\}, \end{cases}$$  \hspace{1cm} (2)$$

where $V_Y$, $V_C$, $V_N$ and $V_{ARG}$ are target vocabularies of process variables, constants, problem variables and augments, $loc(y, T)$ is the location of token $y$ in problem text $T$. (2) Self-limited decoder narrows the search space of output tokens. It limits the output candidates of problem variables $N$ and augments $ARG$ into that appear in the problem text. Concretely, the probability of predicted token $y$ is:

$$P(y) = \text{Softmax}(\text{Score}(h^D, c, e(y))),$$  \hspace{1cm} (3)$$

where $y \in \{V_Y, V_C, V_N \cap T, V_{ARG} \cap T\}$. $\text{Score}$ is the score function, $h^D$ is the hidden output of decoder, $c$ is the context vector generated from $H^E$ using the same attention mechanism as [Bahdanau et al., 2015], $e(y)$ is the token embedding of candidates. In experiments, We find that the self-limited GRU achieves even better performance than complex tree decoders and with much faster training and inference speed.

5 Data Augmentation

Despite that PGPS9K is the largest dataset so far and of high-quality, it still cannot satisfy the model learning of PGPSNet well, especially for the structural and semantic pre-training task. Therefore, we adopt five data augmentation strategies based on diversity and equivalence of geometric representation, taking the problem in Figure 4 as an illustrative example:

- **Token Replacement:** The replaceable tokens include points, angle IDs and augments three types. Once a token is changed, all same tokens in textual clauses and textual problem should be replaced uniformly. Point B is replaced as point V, then get new texts: "line V C D", "CA lieson E V D", "VD ⊥ EA on C", "Find VD".

- **Connection Rotation:** The connection relationship in structural clauses could be re-represented by changing the location order of points. "line B C D" is equivalent to "line D C
**8.5. Representation Transposition**: There are several equivalent representations of geometric primitives of line, angle and arc, e.g., "EA = AE", "\( \angle \text{STR} = \angle \text{RTS} \)", "EF = FE". We randomly transpose the geometric primitives representation in semantic clauses.

**Clauses Shuffle**: We shuffle semantic clauses to produce new ID of problem variable while modify the corresponding solution program. When semantic clauses are adjusted as "CA = 3\( (N_0) \), BD \bot EA on C, EC = 2\( (N_1) \)" , the solution program is changed as "\( \cdot \cdot \cdot \) Gougu N_0 V_1 V_0 \cdot \cdot \cdot \".

**Diagram Flip**: Since the textual content is already parsed in semantic clauses, the visual text in diagram could be ignored. So the flipped or rotated diagram is identical to the original diagram for problem. These five augmentation strategies are independent and can be incorporated with each other. Abundant samples generated from data augmentation equip model with basic geometric knowledge, thus promoting high-level reasoning.

### 6. Experiment

#### 6.1. Setup

**Implementation details.** Our model is implemented using PyTorch on one GTX-RTX GPU. The CNN model adopts the ResNet10 [He et al., 2016], feeding with diagram images resized as 128 \times 128. The language model select a transformer encoder [Vaswani et al., 2017], having 6 layers, 8 attention heads, and a hidden embedding size of 1024. The GRU encoder is a two-layer bidirectional GRU [Cho et al., 2014] with input embedding size 256 and hidden state size 512. The self-limited decoder is a two-layer GRU setting same input embedding size and hidden state size of 512. The random probability of data augmentation is set as 0.7 in pre-training and 0.5 in training. We choose the AdamW optimizer [Loshchilov and Hutter, 2017] with weight decay \( 1e^{-2} \) and step decline schedule with decaying rate 0.5. During pre-training, the learning rate of language model is initialized as \( 5e^{-4} \) decaying at 1K, 2K and 3K epochs with a total 4K epochs. During training, all modules of PGPSNet train together with initial learning rate as \( 5e^{-5} \) for language model and \( 1e^{-3} \) for other modules, decaying at 160, 280, 360, 440 and 500 uniformly with a total 540 epochs. In addition, the batch size and dropout rate are set as 128 and 0.2 in all processes.

**Evaluation metrics.** We evaluate performance of geometric solvers at two levels: numerical answer and solution program. At each level, there are three evaluation patterns: Completion, Choice and Top-3. In Completion, the symbolic solver gives the result searched for and the neural solver selects the first executable solution program. The Choice is defined as choosing the correct option from four candidates but selecting one randomly if answer is not in. The Top-3 computes the accuracy ratio of correct answer existing among top three solution candidates, less strict than Completion pattern. We set beam size to 10 being the same as [Chen et al., 2021] for all evaluation manners. Noting that the performance of solution program is often lower than the actual result due to the equivalence of operation orders and diversity of solution strategies.

**Datasets.** We conduct experiments on PGPS9K dataset split in two ways as introduced in Section 3.1, signified as Geometry3K and PGPS9K in the following experiments. To simplify experiments, we adopt the textual clauses generated from ground truth of diagram annotation for both symbolic solvers and neural solvers. The language model is pre-trained from scratch on our PGPS9K on account of huge gap from natural corpus and short of geometric corpus. Datasets GeoQA [Chen et al., 2021] and GeoQA+ [Cao and Xiao, 2022] are not considered in experiments because they have no fine-grained diagram annotation and their diagrams are low-quality, hard to be parsed by existing diagram parsers.

#### 6.2. Comparison with State-of-the-art Methods

To evaluate our PGPSNet solver, we compare its performance with state-of-the-art GPS solvers. InterGPS [Lu et al., 2021], NGS [Chen et al., 2021] and Geoformer [Chen et al., 2022].

**Symbolic solvers.** The InterGPS is a rule-based symbolic solver which parses the problem text and diagram into formal language. Table 2 displays the performances of InterGPS in three input modes with the best search strategy, where Inter-GPS(Predict) indicates that diagram and text formal language are predicted by its diagram and text parser, Inter-GPS(Diagram GT) indicates that text formal language is generated by its text parser but diagram formal language uses the ground truth, Inter-GPS(All GT) indicates that all utilize the ground truth. On Geometry3K, the results show that our PGPSNet outperforms Inter-GPS(Predict), achieves comparable performance as Inter-GPS(Diagram GT), and is inferior to InterGPS(All GT) in Completion. But in Choice, PGPSNet has
surpassed all input modes and even gains a 2.0% improvement over InterGPS(All GT). On PGPS9K, we implement the InterGPS(Diagram GT) by converting diagram annotation into diagram formal language and using its text parser. The PGPSNet shows more performance improvements in Completion and Choice, and result in Top-3 implies the improvement potential of our PGPSNet.

Neural solvers. NGS and Geoformer are two neural solvers and we re-implement them on our dataset. For full and fair comparison, we keep their pre-trained diagram encoder fixed, take both semantic clauses and textual problem as their text input, and employ the same data augmentation. Thanks to our good modal representation and effective modal fusion, our PGPSNet demonstrate superior performance improvements compared to baseline neural solver, NGS and Geoformer as displayed in Table 2. Besides, we note that there remains a huge performance gap between our solver and human expert, having much room for improvement.

### 6.3 Ablation Study

To demonstrate effects of different modules of PGPSNet, we conduct ablation studies on Geometry3K, taking the self-limited decoder, data augmentation, structural clauses and pre-training as ablation objects as displayed in Table 3. The comparison between row 1 and row 4 shows that the data augmentation promotes GPS through adding geometric representation knowledge into the augmented data. By comparing row 2 to row 4, we find that the self-limited decoder improves performance of geometric reasoning, who limits feature and search space to reduce difficulty of model learning. The language model, with structural and semantic pre-training, brings an amazing performance gain, especially in Completion with a 26.6% answer accuracy improvement, as shown in row 4 and row 6. We also discover that structural clauses show less affect without pre-training compared row 3 with row 4, but obtain a remarkable improvement with pre-training as displayed in row 5 and row 6, revealing that basic connection relations can promote model’s structure cognition by a befitting modal fusion approach. The performance trends of solution program are consistent with the numerical answer in all experiments.

### 6.4 Case Study

We also conduct case studies for discussing the ability and limitation of our PGPSNet as shown in Figure 5. The case (a) examines the application of Angle Bisector Theorem. The methods of NGS and PGPSNet w/o LM fail to determine the relation of corresponding edges correctly while our PGPSNet取得 the right solution program. The case (b) is challenging and need to combine two conditions of Segment Length Theorem. Solutions of all solvers are incorrect for case (b) but the PGPSNet’s is most similar to the ground truth with only wrong in the second step, illustrating that PGPSNet is not yet qualified for complex geometric reasoning nowadays but has great potential.

### 7 Conclusion

In this work, we propose a new diagram-text fusion solver PGPSNet combining textual clauses parsed from diagram, and construct a large-scale and fine-annotated GPS dataset PGPS9K. Benefiting from effective modal representation and efficient modal fusion, PGPSNet makes full use of basic structural and semantic information to implement geometric reasoning. Besides, the interpretable solution program and well-designed data augmentations provide model with critical geometric knowledge for GPS such as geometry theorem and geometric representation. The experimental results demonstrate the potential of neural solvers and our work still has large room for improvement with more training samples or more elaborate modal fusion.

<table>
<thead>
<tr>
<th>Self-limited Decoder</th>
<th>Data Aug</th>
<th>Structural Clauses</th>
<th>Pre-trained LM</th>
<th>Ans acc</th>
<th>Prog acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>32.5</td>
<td>27.2</td>
</tr>
<tr>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>28.2</td>
<td>25.4</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>36.6</td>
<td>33.9</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>38.4</td>
<td>34.8</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>48.1</td>
<td>45.4</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>65.0</td>
<td>62.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Ans acc</th>
<th>Prog acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>NGS</td>
<td>65.2</td>
<td>53.1</td>
</tr>
<tr>
<td>PGPSNet &amp; GT</td>
<td>57.6</td>
<td>45.6</td>
</tr>
<tr>
<td>PGPSNet w/o LM</td>
<td>50.7</td>
<td>58.6</td>
</tr>
<tr>
<td>GT</td>
<td>62.8</td>
<td>59.2</td>
</tr>
</tbody>
</table>

Table 3: Ablation studies on Geometry3K.

Figure 5: Case studies on PGPS9K.
Acknowledgments
This work has been supported by the National Key Research and Development Program under Grant No. 2020AAA0109700, the National Natural Science Foundation of China (NSFC) grants U20A20223 and 61721004.

References


[Lu et al., 2021] Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. Inter-GPS: Interpretable geometry problem solving with formal language and symbolic reasoning. In ACL, 2021.


