Plane Geometry Diagram Parsing

Ming-Liang Zhang[1,2,*] , **Fei Yin**[1,2] , **Yi-Han Hao**[1,3] and **Cheng-Lin Liu**[1,2]

[1]National Laboratory of Pattern Recognition, Institute of Automation of Chinese Academy of Sciences
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences
[3]School of Electronic Information Engineering, Beijing Jiaotong University
zhangmingliang2018@ia.ac.cn, fyin@nlpr.ia.ac.cn, 20120004@bjtu.edu.cn, liucl@nlpr.ia.ac.cn

Abstract

Geometry diagram parsing plays a key role in geometry problem solving, wherein the primitive extraction and relation parsing remain challenging due to the complex layout and between-primitive relationship. In this paper, we propose a powerful diagram parser based on deep learning and graph reasoning. Specifically, a modified instance segmentation method is proposed to extract geometric primitives, and the graph neural network (GNN) is leveraged to realize relation parsing and primitive classification incorporating geometric features and prior knowledge. All the modules are integrated into an end-to-end model called PGDP-Net to perform all the sub-tasks simultaneously. In addition, we build a new large-scale geometry diagram dataset named PGDP5K with primitive level annotations. Experiments on PGDP5K and an existing dataset IMP-Geometry3K show that our model outperforms state-of-the-art methods in four sub-tasks remarkably. Our code, dataset and appendix material are available at https://github.com/mingliangzhang2018/PGDP.

1 Introduction

Automatic geometry problem solving is a long-standing problem and has important applications in the intelligent education field [Chou *et al.*, 1996; Seo *et al.*, 2015; Amini *et al.*, 2019]. The problem involves text parsing, corresponding diagram parsing and logical reasoning. Previous research works [Sachan *et al.*, 2017; Sachan *et al.*, 2020] mainly concentrated on text parsing and logical reasoning, but little attention has been paid to diagram parsing [Seo *et al.*, 2014; Lu *et al.*, 2021]. Geometry diagrams carry rich information about the geometry problem, which can provide crucial cues to aid problem solving. In this work, we focus on plane geometry diagram parsing (PGDP) and propose a powerful diagram parser.

Generally, the PGDP task involves identifying and locating visual primitives in the diagram and discovering relationships among them. As shown in Figure 1, a geometry diagram consists of various types and layouts of geometry, symbols and
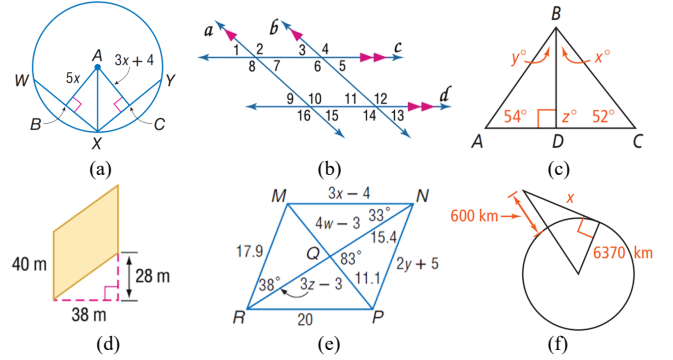

Figure 1: Examples of plane geometry diagram.

texts, and these visual primitives are semantically related to each other in various ways. Due to the diversity of style and the interference of primitives, traditional methods, such as Hough transform and Freeman chain-code [Pratt, 2007], perform poorly in geometric primitive extraction. Meanwhile, the spatial, structural and semantic relations among primitives cannot be parsed correctly by simple rule-based methods [Seo *et al.*, 2014; Lu *et al.*, 2021]. Therefore, great efforts are needed for geometric primitive extraction and between-primitive relationship parsing.

We cast geometric primitive extraction as an instance segmentation problem. Geometric primitives such as lines and arcs are often slender and overlapped. Thus, bounding box based instance segmentation methods [He *et al.*, 2018; Neven *et al.*, 2019; Ying *et al.*, 2021] are not suitable for this task. Our proposed PGDP framework instead employs a geometric segmentation module (GSM), consisting of a semantic segmentation branch and a segmentation embedding branch, to cluster multi-class primitive instances at pixel level, so as to overcome the issues stated above.

For primitive relation parsing, we model PGDP as a special scene graph generation (SGG) problem [Xu *et al.*, 2017; Liu *et al.*, 2021]. In contrast to ordinary SGG, as shown in Figure 2, PGDP deals with graphs with heterogeneous nodes and multi-edges associated, when primitives and their relations are seen as nodes and edges, respectively. To optimize reasoning incorporating geometry prior knowledge, we adopt a GNN module (GM) aggregated with visual, spatial
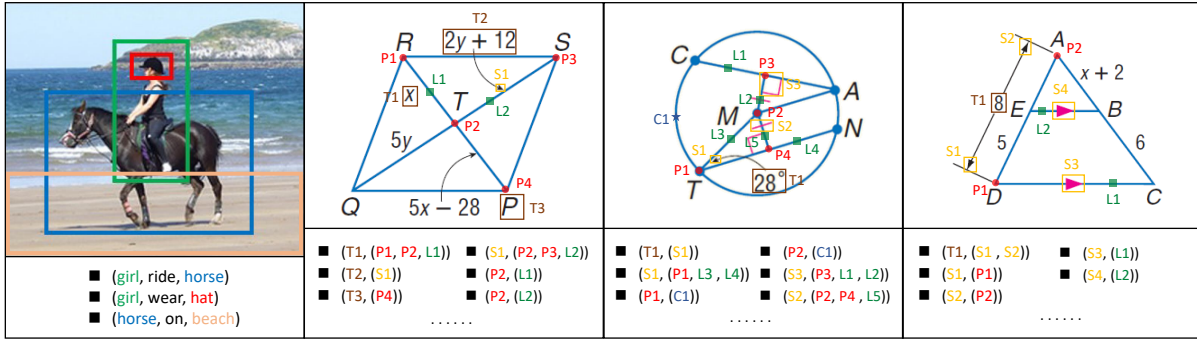
---
*Contact Author

Figure 2: Comparison between tasks of SGG (first image) and PGDP (next three images). Relation tuples are shown below each image. 'P#','L#','C#','T#' and 'S#' denote instances of point, line, circle, text and symbol, respectively.

and structural information to predict the primitive relation and identify the text class simultaneously.

Integrating with GSM and GM, we present the deep learning model for PGDP called PGDPNet. The PGDPNet is trained end-to-end so as to optimize the overall primitive extraction and relation reasoning performance. Also, to facilitate the research of PGDP, we build a new large-scale geometry diagram dataset named PGDP5K, labeled with annotations of primitive locations, classes and their relations. Experiments on PGDP5K and an existing dataset IMP-Geometry3K demonstrate that our method can boost the performance of primitive detection, relation parsing and geometry formal language generation prominently, compared to state-of-the-art methods, and consequently improves the accuracy of geometry problem solving.

The contributions of this work are summarized in three folds: (1) We propose the PGDPNet, the first end-to-end deep learning model for explicit geometry diagram parsing. (2) We build a large-scale dataset PGDP5K, containing fine-grained annotations of primitives and relations. (3) Our method demonstrates superior performance of geometry diagram parsing, outperforming previous methods significantly.

2 Related Work

Automatic analysis of geometry diagrams has been studied in two main aspects: primitive extraction and relation reasoning. As to primitive extraction, traditional methods such as Hough transform and its improved methods [Pratt, 2007] are still adopted in most recent geometry diagram parsing works [Seo *et al.*, 2014; Seo *et al.*, 2015; Gan *et al.*, 2018; Lu *et al.*, 2021] for their simplicity and efficiency. However, in the scenes of complex layout and multi-primitive interference, traditional methods inevitably suffer severe performance degradation. Deep learning based geometric primitive extraction methods [Huang *et al.*, 2018; Zhou *et al.*, 2019] have been proposed recently. Nevertheless, they only focus on one type of geometric primitive such as straight line in nature scenes. Research works about geometric relation reasoning of diagrams are undergoing. Some methods [Seo *et al.*, 2014; Lu *et al.*, 2021] use greedy or optimization strategies based on distance and content rules, but cannot parse complicated between-primitive relations correctly. Our work, inspired by the SGG task [Xu *et al.*, 2017; Liu *et al.*, 2021;

Guo *et al.*, 2021], reasons primitive relations with the GNN model [Veličković *et al.*, 2018; Ye *et al.*, 2020]. The detailed comparison between these two tasks will be described in Section 3.2. To sum up, our work proposes a more powerful geometry diagram parser with a novel and effective scheme for diagram primitive extraction and reasoning.

3 Preliminary

Before the description of problem formulation and solution, we introduce the terms involved in PGDP. The basic element in plane geometry diagram is called *primitive*, which is generally categorized into *geometric primitive* and *non-geometric primitive*. The main categories of geometric primitives are *point*, *line* and *circle* (*arc*), while non-geometric primitives include *text* and *symbol*. *Predicate* is the general term of geometric shape entity, geometric relation or arithmetic function. *Proposition* is the logical expression combined with predicate and primitive. A set of propositions makes up the *geometry formal language* of a diagram.

3.1 Task Formulation

The PGDP consists of three fundamental sub-tasks: (1) detection and identification of primitives; (2) building basic relationships among primitives; (3) generating the geometry formal language. In this work, we model PGDP as special SGG, which is formulated in the following form:

$$
\begin{aligned}
O &= \{O_{geo}, O_{sym}, O_{text}\}, \\
B &= \{B_{geo}(\text{mask}), B_{sym}(\text{box}), B_{text}(\text{box})\}, \quad (1) \\
R &= \{R_{geo2geo}, R_{text2geo}, R_{sym2geo}, R_{text2sym}\},
\end{aligned}
$$

where O is the primitive set, the subscripts geo, sym, text stand for geometric primitive, symbol and text, respectively; B is the primitive position set (geometric and non-geometric primitives are represented by mask and bounding box); R is the primitive relation set among geometric primitive, symbol and text. $G=\{O,B,R\}$ constitutes a special scene graph. The parsing on image I is aimed to predict the constituents of G:

$$
\begin{aligned}
\mathcal{P}(G \mid I, K) = {}&\mathcal{P}(B \mid I) \cdot \mathcal{P}(\{O_{geo}, O_{sym}\} \mid I, B) \\
&\cdot \mathcal{P}(R, O_{text} \mid \{O_{geo}, O_{sym}\}, B, I, K),
\end{aligned} \quad (2)
$$

where K denotes the knowledge graph of geometry; $\mathcal{P}(B \mid I) \cdot \mathcal{P}(\{O_{geo}, O_{sym}\} \mid I, B)$ refers to the steps of object detection and instance segmentation for obtaining the
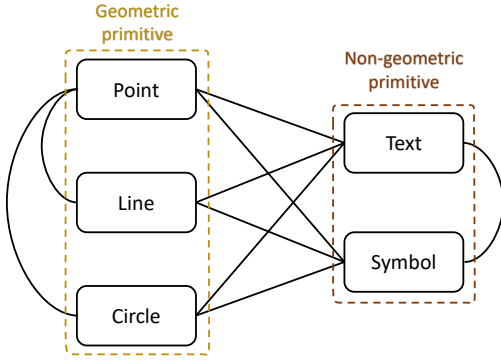
Figure 3: Primitive relationship graph of plane geometry diagram.

primitive positions and classes of symbol and geometric primitive; $\mathcal{P}\left(R, O_{text} \mid \{O_{geo}, O_{sym}\}, B, I, K\right)$ stands for relation inference to acquire the primitive relationship and text class. At last, we generate the proposition set to form the readable description language.

3.2 Comparison with SGG

PGDP is different from general SGG in two respects as shown in Figure 2. First, SGG obtains coarse box positions of targets from nature scene images through object detection, while PGDP is aimed to get the fine-grained masks of geometric primitives through instance segmentation, because they largely overlap with each other in box. Second, the SGG constructs the relationship graph in the form of subject-predicate-object triplets, which has no necessary dependency between the predicate and subject/object classes, while the between-object relationship in PGDP is mostly geometric, and specific relations (predicate) could be inferred according to primitive classes (subject/object) and prior knowledge, do not require an extra classification step.

4 PGDP5K Dataset

Although several datasets [Seo *et al.*, 2015; Sachan *et al.*, 2017; Lu *et al.*, 2021; Chen *et al.*, 2021] for solving geometry problems have been proposed, there is no dataset focusing on PGDP. To facilitate research in geometry problem solving, we build a new large-scale and fine-annotated plane geometry diagram dataset named PGDP5K[1].

4.1 Statistics

The PGDP5K dataset contains 5,000 diagram samples, consisting of 1,813 non-duplicated images from the Geometry3K dataset and other 3,187 images collected from three popular textbooks across grades 6-12 on mathematics curriculum websites[2]. We randomly split the dataset into three subsets: train set (3,500), validation set (500) and test set (1,000). In contrast to previous datasets, diagrams in PGDP5K have more complex layouts such as multiple classes of primitives and complicated primitive relations, which make our dataset more challenging. Specifically, we divide the geometric primitive, text and symbol into 3, 6 and 16 classes respectively, and Appendix A displays several instances of each

class primitive. Some classes of primitives have great within-class style variations. The Appendix B shows distributions of shape, symbol, text and relation.

4.2 Annotations

The annotations of PGDP5K dataset include three types: geometric primitive, non-geometric primitive and primitive relation. These annotations can generate geometry formal language automatically and uniquely. As to geometric primitives, we annotate their pixel positions and uniform pixel widths. For non-geometric primitives, bounding box, symbol class, text class and text content are labeled. As to primitive relations, we construct a relation graph of elementary relationships among primitives exhibited in Figure 3, where we only construct relations between point and line, point and circle for relations of geometric primitives, because other high-level relations among geometric primitives can be derived from these two basic relations. A two-tuple with multiple entities is used to represent one relationship as demonstrated in Figure 2. Compared with the triplet of SGG, we take point, symbol and text as subjects, and serve other related primitives as objects, neglecting the relation class term. Finally, we define geometry proposition templates of basic relations listed in Appendix C. For more annotation details, please refer to the website of PGDP5K dataset. In addition, we re-annotate diagrams of the Geometry3K in our way and rename it IMP-Geometry3K.

5 Model

The proposed PGDPNet depicted in Figure 4 is presented hereon in detail, focusing on the geometric primitive segmentation and the primitive relation parsing.

5.1 Backbone Module (BM)

A typical FPN architecture [Lin *et al.*, 2017] is used as the BM for visual feature extraction. The FPN layers P3-P7 are exploited for text and symbol detection, and the FPN layer P2 embedded with the location maps is shared by the geometric segmentation module (GSM) and the visual-location embedding module (VLEM). Visual features mixed with spatial information will facilitate model learning of follow-up tasks.

5.2 Non-geometric Detection Module (NDM)

The non-geometric primitives, symbol and text, are detected in NDM. Given the diverse size scales of text and symbol in geometry diagrams, an anchor-free detection method FCOS [Tian *et al.*, 2020] is utilized to avoid the setting of prior anchors and improve the detection speed. This module consists of regression, center-ness and classification branches. The loss in training is $L_{FCOS} = L_{reg} + L_{cns} + L_{cls}$, where L_{reg}, L_{cns} and L_{cls} are the losses of three branches.

5.3 Geometric Segmentation Module (GSM)

Due to complex layouts and elongated shapes, traditional methods and current instance segmentation approaches based on bounding boxes are all not suitable for geometric primitive extraction. We propose a new instance segmentation method, where two branches, semantic segmentation branch and segmentation embedding branch, together implement the multi-class instance segmentation of geometric primitives. The semantic segmentation branch performs binary segmentation

---

[1] http://www.nlpr.ia.ac.cn/databases/CASIA-PGDP5K
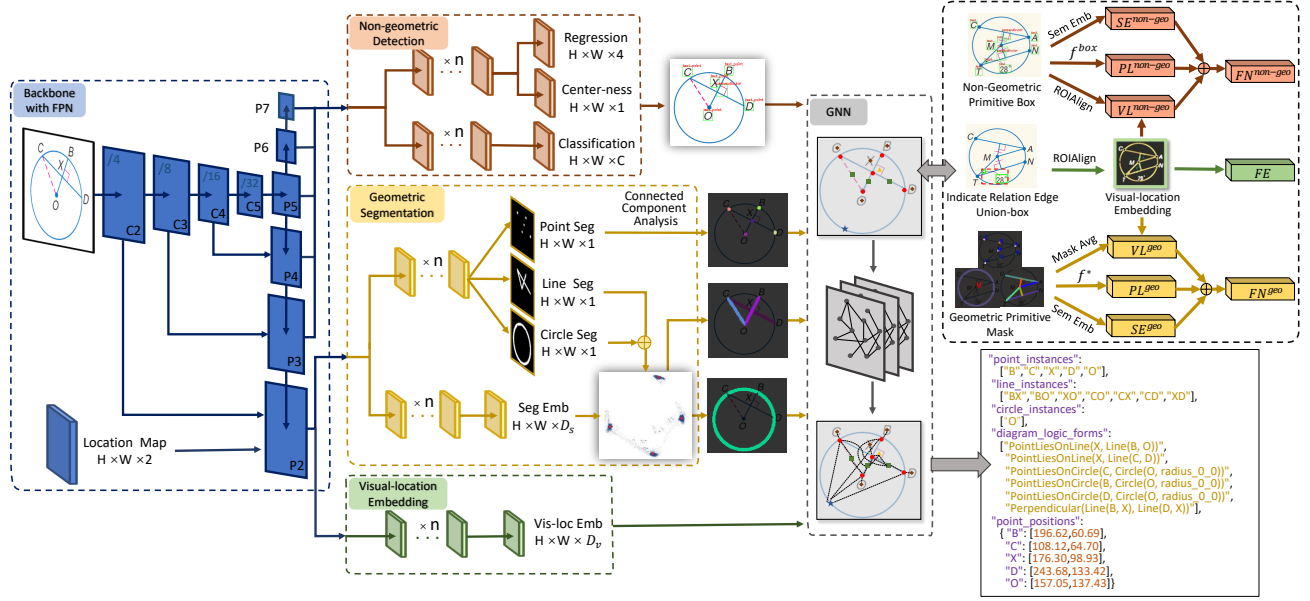
[2] https://www.mheducation.com/

Figure 4: Overview of our proposed PGDPNet.

with the weighted binary cross-entropy (BCE) loss:

$$L_{bs*} = \frac{-w_*}{M_{map}} \sum_{i=1}^{M_{map}} y_i^* \log(p_i^*) + (1-y_i^*) \log(1-p_i^*), \quad (3)$$

where * denotes the primitive class, w is the weight ratio for balancing positive and negative class pixels, empirically as $w_p = 5, w_l = 1, w_c = 4$, and M_{map} is the pixel number of segmentation map. Then the segmentation loss of all classes is $L_{bs} = L_{bsp} + L_{bsl} + L_{bsc}$. The discriminative loss [De Brabandere *et al.*, 2017; Neven *et al.*, 2018] is used in the segmentation embedding branch to better differentiate instances:

$$\begin{cases} L_{dist} = \dfrac{1}{N_{lc}(N_{lc}-1)} \sum\limits_{n_1=1}^{N_{lc}} \sum\limits_{\substack{n_2=1 \\ n_2 \neq n_1}}^{N_{lc}} [2\delta_d - \|\mu_{n_1} - \mu_{n_2}\|]_+^2, \\ L_{var} = \dfrac{1}{N_{lc}} \sum\limits_{n=1}^{N_{lc}} \dfrac{1}{M_n} \sum\limits_{i=1}^{M_n} [\|\mu_n - x_i\| - \delta_v]_+^2, \end{cases} \quad (4)$$

where $N_{lc} = N_l + N_c$ is the sum of instance number of line and circle, M is the pixel instance number, x is the pixel embedding, μ is the center of instance embedding. In default, the threshold δ_d of center distance is set as 1.5 and threshold δ_v of embedding radius is set as 0.5. The line instances and circle instances are learned simultaneously to obtain more distinctive features. In contrast, due to inherent instance separation in space, point instances are acquired just by connected component analysis according to the results of semantic segmentation, reducing difficulty of model learning. Eventually, the whole loss of GSM is $L_{ins} = L_{bs} + L_{dist} + L_{var}$.

5.4 GNN Module (GM)

After obtaining the primitives, the relationship among primitives is reasoned by the GM. Before that, the VLEM unifies all primitive features and works as the initialization of GM. We treat primitives as nodes and primitive relations as

edges to compose a primitive relation graph, represented as $G = \{V = \{V^{geo}, V^{Non-geo}\}, E = \{e_{ij}|K\}\}$, where V^{geo}, $V^{Non-geo}$ and E denote the node sets of geometric and non-geometric primitive, and the edge set. The cardinality of geometric node set is $|V^{geo}| = N_p + N_l + N_c$, where N_p is the instance number of point. The original relation graph is a difficult hyper-graph with heterogeneous nodes (mask and box) and multi-edge relationship. For efficient solution, we transform the graph to a simple isomorphic graph and then construct the sparse graph like the one in Figure 3. Specifically, we only connect nodes that may have relations according to geometric prior knowledge K, and then categorize edges to determine final results.

The initial features of nodes are represented by the fusion of visual-location embedding VL, parsing position feature PL and class semantic feature SE. For visual-location embedding, we transform mask features of variable shapes into vector features of fixed length by the mask average:

$$VL_i^{geo} = B^T \frac{\text{mask}_i}{\|\text{mask}_i\|_1}, \quad (5)$$

where B is the visual-location embedding map. The mask average can also reduce the influence of error caused by inaccurate segmentation. In addition, we employ the RoIAlign method [He *et al.*, 2018] to normalize multi-scale box features as vector features of the same length:

$$VL_i^{non-geo} = \text{RoIAlign}(B, \text{box}_i). \quad (6)$$

Besides, parsing position in plane space is a significant feature of geometric primitives. We incorporate it into the GM to further facilitate relation building, formulated as:

$$PL_i = f^*(pr_i^*), \quad pr^* = \begin{cases} [x, y], & * = \text{point}, \\ [x_1, y_1, x_2, y_2], & * = \text{line}, \\ [x, y, r], & * = \text{circle}, \\ [x_1, y_1, x_2, y_2], & * = \text{box}, \end{cases} \quad (7)$$

	InterGPS	PGDPNet w/o GNN	PGDPNet
Non-geometric Primitive Detection	Object Detection	Object Detection	Object Detection
Geometric Primitive Detection	Hough Transform	Instance Segmentation	Instance Segmentation
Text Classification	Content Rules	Detection Classification	Joint Classification
Relation parsing	Distance Rules	Distance Rules	GNN
Has Parsing Position	Yes	No (localizable)	No (localizable)
Is End-to-End	No	No	Yes

Table 1: Functions of comparison methods.

where pr^* is the parsing representation of primitives, while point, line, circle and box denotes as a point, two endpoints, center with radius, and top-left point with bottom-right point, respectively, f^* is a network module of two fully-connected layers with ReLU activation. In that way, the final node feature is formulated as $FN_i^\& = VL_i^\& + PL_i^\& + SE_i^\&, i=1\cdots|V^\&|$, where $\&$ is the primitive class geo or $non-geo$.

The edge features are only generated for arrow indication relations, and the rest are aggregated through layer propagation of GNN. Because of elongated shape, the detection performance of arrows is less satisfactory. Instead of detecting boundary boxes of arrows, we use the union box of head box and corresponding text box to represent the relation:

$$FE_{ij} = \begin{cases} \text{RoIAlign}(B, \text{box}_i \cup \text{box}_j), & (v_i, v_j) = (\text{head}, \text{text}), \\ \mathbf{0}, & \text{others}. \end{cases} \quad (8)$$

Although union box cannot enclose the whole arrow in some cases, it works well in experiments because the feature box has a larger receptive field than its own size with the FPN.

The GM performs two sub-tasks. The first is predicting the edge class to judge whether existing relationship between nodes, with the loss function as:

$$L_{edge} = \frac{-1}{|E|} \sum_{i=1}^{|E|} y_i \log(p_i) + (1-y_i) \log(1-p_i). \quad (9)$$

The second is the fine-grained text classification with CE loss:

$$L_{node} = \frac{-1}{N_t} \sum_{i=1}^{N_t} \sum_{c=1}^{C_{text}} y_{ic} log(p_{ic}), \quad (10)$$

where N_t is the instance number of text and C_{text} is the text class number. Compared with visual features alone, the combined features including spatial structure information promote fine-grained classification, considering that some texts of different classes are visually identical. As to the architecture of GNN, the edge graph attention network (EGAT) [Guo et al., 2021; Ye et al., 2020] is employed as the backbone of GM for its excellent reasoning ability among nodes and edges, and the whole loss of GM is $L_{GNN} = L_{edge} + L_{node}$.

5.5 Training and Testing

During the training, the model PGDPNet is trained end-to-end with the aggregated loss:

$$L_{all} = L_{FCOS} + \alpha \cdot L_{ins} + \beta \cdot L_{GNN}. \quad (11)$$

Empirically we set the weight coefficients $\alpha = \beta = 4$. During testing, according to binary masks obtained from the semantic segmentation branch of GSM, the segmentation embedding branch clusters embedding features to get instances of

line and circle by the MeanShift cluster method. The parsing position of instance masks could be located accurately by simple fitting methods due to precise segmentation. The extracted geometric and non-geometric primitives go through the VLEM to generate initial features of GM, then the GM gets relations among primitives via node and edge classification. In the end, geometric propositions are produced according to geometry prior knowledge and language grammar.

6 Experiments

6.1 Experimental Setup

Implementation Details

We implemented our method using the PyTorch and FCOS framework [Tian et al., 2020]. The backbone adopts MobileNetV2 [Sandler et al., 2018]. The NDM, GSM and VLEM all use 3 groups of 128-channel convolution layers with corresponding BatchNorm layers. The segmentation embedding dimensionality is 8 and the visual-location embedding dimensionality is 64. The layer number of GM is 5 and the feature dimensionalities of nodes and edges are all set to 64. To improve the diversity of samples, two enhancement strategies, random scale scaling and random flipping, are exploited during the training. We choose the Adam optimizer with an initial learning rate $5e^{-4}$, weight decay $1e^{-4}$, step decline schedule decaying with a rate of 0.2 at 20K, 30K and 35K iterations. We train our model in 40K iterations with batch size of 12 on 4 TITAN-Xp GPUs.

Comparison Methods

To evaluate the effects of different modules, we compare three methods: InterGPS, PGDPNet without GNN, and PGDPNet. As described in Table 1, these approaches respectively adopt different technologies on sub-tasks, where our PGDPNet is a concise and efficient framework that could be learned end-to-end from datasets.

Evaluation Protocols

We evaluate the methods at four levels: primitive detection, relation parsing, geometry formal language generation, and problem solving. Considering that some labels of bounding box are loose especially for single-word texts and arrowheads, we set threshold IOU=0.5 to evaluate the non-geometric primitive detection. As to geometric primitive extraction, there are two evaluation manners: one (**manner 1**) is parsing position evaluation that applies to the Hough transform route and the other (**manner 2**) is mask evaluation designed for the instance segmentation route. We set the distance threshold as 15 consistent with the InterGPS [Lu et al., 2021] for the first manner and set IOU as 0.75 by de-

		Freeman	GEOS	PGDPNet
Point	Precision	67.46	76.51	**99.65**
	Recall	80.41	93.44	**99.71**
	F1	73.37	84.13	**99.68**
Line	Precision	50.78	66.99	**99.30**
	Recall	80.43	90.46	**99.51**
	F1	62.25	76.98	**99.40**
Circle	Precision	90.72	98.25	**99.85**
	Recall	97.75	99.24	**99.96**
	F1	94.10	98.74	**99.90**

Table 2: Detection performance of geometric primitives with the evaluation manner 1.

fault for the second manner. As to relation parsing, we divide one multivariate relation into multiple binary relations, and evaluate the precision, recall and F1 of binary relation terms. The geometry formal language is characterized by diversity and equivalence, for example, "Angle(P,R,Q)" is equivalent to "Angle(P,R,N)" in Figure 1(e). For rationality and fairness of evaluation, we improve the existing evaluation method [Lu *et al.*, 2021] focusing on propositions with line and angle. Experimental results are evaluated on four indicators: Likely Same (F1$\geq$50%), Almost Same (F1$\geq$75%), Perfect Recall (recall=100%) and Totally Same (F1=100%).

6.2 Primitive Detection
To evaluate the effects of NDM and GSM, we give the performance of primitive extraction on PGDP5K. Table 2 depicts the geometric primitive detection results using the evaluation manner 1, in which line instance covers all collinear line segments. We can see that our approach achieves a remarkable improvement over traditional methods such as Freeman [Pratt, 2007] and GEOS [Seo *et al.*, 2015], particularly on point and line. Appendix D lists the performance of all primitive classes adopting the evaluation manner 2. We find that most primitives could be well located and recognized except some minority classes, and joint classification in the GNN evidently improves the performance of text recognition compared with classification in the NDM.

6.3 Primitive Relation Parsing
To better demonstrate advantages of graph feature generation, we conducted ablation studies in primitive relation parsing. Table 3 displays performances of different feature initialization methods, where baseline denotes the method of with only visual-location embedding, SE and PL refer to class semantic feature and parsing position feature formulated in Eq (7), respectively. The performance gap is mainly reflected in the relationships of text2geo, sym2geo and text2head. However, most relations among primitives belong to geo2geo, so the overall performance of relationships shows little difference. We also compare the methods using another evaluation indicator *complete accuracy*, which refers to the proportion of complete correct sample. On the whole, fusion with parsing position and class semantic information makes model easier to learn representative features so as to promote relation reasoning, and gains 1.7% complete accuracy improvement.

		Baseline	w SE	w PL	w SE&PL
All	Precision	98.96	99.16	99.10	**99.16**
	Recall	96.97	97.01	**97.11**	97.07
	F1	97.96	98.07	98.08	**98.11**
Geo2Geo	Precision	98.84	**99.15**	99.09	99.13
	Recall	98.53	98.56	**98.69**	98.60
	F1	98.68	98.85	**98.89**	98.86
Text2Geo	Precision	99.09	99.27	**99.38**	99.27
	Recall	96.16	96.61	96.36	**96.84**
	F1	97.60	97.94	97.84	**98.04**
Sym2Geo	Precision	99.06	98.71	99.01	**99.07**
	Recall	94.13	94.89	95.02	**95.27**
	F1	96.53	96.76	96.97	**97.13**
Text2Head	Precision	97.70	98.03	98.03	**98.08**
	Recall	91.95	92.26	92.57	**95.05**
	F1	94.74	95.06	95.22	**96.54**
Complete Acc		81.50	82.50	82.60	**83.20**

Table 3: Ablation studies of primitive relation parsing. "A2B" denotes the relationship between class A and class B by default.

		IMP-Geometry3K	PGDP5K
All	Likely Same	73.71 / 99.17 / **99.33**	65.70 / 98.40 / **99.00**
	Almost Same	50.08 / 95.51 / **98.50**	44.40 / 93.10 / **96.60**
	Perfect Recall	45.26 / 81.03 / **92.18**	40.00 / 79.70 / **86.20**
	Totally Same	34.28 / 80.53 / **91.51**	27.30 / 78.20 / **84.70**
Geo2Geo	Likely Same	69.88 / **99.67** / 99.50	63.90 / **99.10** / 99.00
	Almost Same	56.24 / **99.50** / 99.00	49.40 / **97.30** / 97.10
	Perfect Recall	74.71 / **99.33** / 99.17	78.70 / 96.90 / **97.40**
	Totally Same	47.59 / **98.84** / 98.33	40.80 / 93.60 / **94.50**
Non-geo 2Geo	Likely Same	77.04 / 96.01 / **99.00**	67.30 / 95.80 / **98.00**
	Almost Same	59.07 / 89.35 / **96.01**	49.80 / 88.20 / **94.90**
	Perfect Recall	50.92 / 81.20 / **92.85**	45.70 / 81.30 / **87.00**
	Totally Same	48.59 / 80.87 / **92.85**	40.50 / 80.60 / **86.40**

Table 4: Evaluation results of specification generation in geometry formal language. "&/&/&" denotes performances of three methods compared: InterGPS, PGDPNet without GNN and PGDPNet.

6.4 Geometry Formal Language Generation
We also conducted experiments in the generation of geometry formal language for a more advanced evaluation. The outcomes of geometry formal language depend on the complete results of primitive extraction and relation reasoning to form comprehensible geometric propositions, and any error in preceding sub-tasks will influence the final generation results. Table 4 shows experimental results of all, geo2geo and non-geo2geo relationship. Our method without GNN sharply outperforms the InterGPS on geo2geo relationship, and the GNN module further improves non-geo2geo relationship reasoning in spite of a slight performance decline on geo2geo, because segmentation results are already accurate enough to determine the geo2geo relation by the distance rules. Finally, on two datasets, our PGDPNet respectively achieves 57.2% and 57.4% improvements of Totally Same of All compared with the InterGPS, and exceeds the one without GNN by 11.0% and 6.5%.

	Text InterGPS	Text GT
Diagram w/o	25.4±0.0	25.4±0.0
Diagram InterGPS	57.5±0.2	58.0±1.7
Diagram PGDPNet w/o GNN	69.3±0.2	70.0±0.4
Diagram PGDPNet	74.1±0.2	74.3±0.3
Diagram GT	**75.9±0.2**	**76.0±0.4**

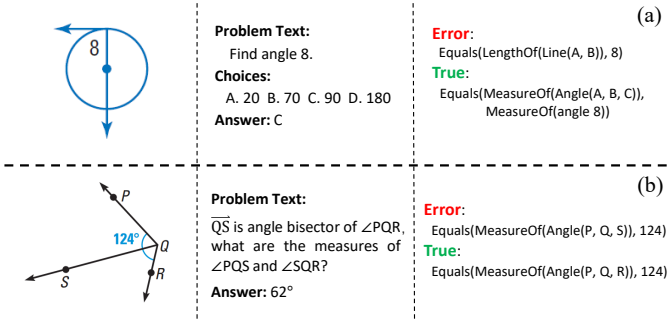Table 5: Problem solving accuracy of InterGPS system on IMP-Geometry3K dataset.

Figure 5: Failure examples of our method.

6.5 InterGPS System Problem Solving

To show the potential of our approach in geometry problem solving, we evaluate the performance using an existing problem solver, the one of InterGPS system, by replacing its geometry diagram parser with ours while remaining other modules unchanged. Table 5 reports the Inter-GPS performance feeding with different sources of propositions. When using the text parser of InterGPS with propositions generated from our PGDPNet, Inter-GPS achieves accuracy of 74.1%, nearly 16.6% higher than the diagram parser of InterGPS. The GM improves performance by 4.8% compared to the one without GNN. Slight gaps among generated diagram propositions, generated text propositions and annotations show that the symbolic geometry solver of InterGPS still has much room to improve.

6.6 Limitations

We show some failure cases of our method in Figure 5. In Figure 5(a), the text "8" is mistaken as the radius of circle, while the problem text shows that it is an angle label. In Figure 5(b), The text "124°" is incorrectly denoted as the degree of $\angle PQS$ but is actually the degree of $\angle PQR$. This reveals that geometry diagram parsing should not rely on images alone but also make full use of textual semantics, and it even involves geometry logical reasoning. Future works will consider incorporating the text description to aid diagram parsing to further improve parsing performance.

7 Conclusion

We propose the first end-to-end deep learning model PGDP-Net for PGDP, which gives explicit primitive instance extraction, classification and between-primitive relationship reasoning. We also construct a new large-scale geometry diagram dataset PGDP5K with primitive level annotations. Experi-

mental results demonstrate the superiority of proposed parsing method. This work promotes the benchmark of plane geometry diagram parsing, and provides a powerful tool to aid geometry problem solving and Q&A.

References

[Amini *et al.*, 2019] Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. MathQA: Towards interpretable math word problem solving with operation-based formalisms. In *NAACL HLT*, 2019.

[Chen *et al.*, 2021] Jiaqi Chen, Jianheng Tang, Jinghui Qin, Xiaodan Liang, Lingbo Liu, Eric Xing, and Liang Lin. GeoQA: A geometric question answering benchmark towards multimodal numerical reasoning. In *Findings of ACL*, 2021.

[Chou *et al.*, 1996] Shang Ching Chou, Xiao Shan Gao, and Jing Zhong Zhang. Automated generation of readable proofs with geometric invariants: II. theorem proving with full-angles. *Journal of Automated Reasoning*, 17:349–370, 1996.

[De Brabandere *et al.*, 2017] Bert De Brabandere, Davy Neven, and Luc Van Gool. Semantic instance segmentation with a discriminative loss function. In *CoRR*, volume abs/1708.0, 2017.

[Gan *et al.*, 2018] Wenbin Gan, Xinguo Yu, Chao Sun, Bin He, and Mingshu Wang. Understanding plane geometry problems by integrating relations extracted from text and diagram. *PSIVT 2017: Image and Video Technology, LNCS*, 10749:366–381, 2018.

[Guo *et al.*, 2021] Yunfei Guo, Wei Feng, Fei Yin, Tao Xue, Shuqi Mei, and Cheng-Lin Liu. Learning to understand traffic signs. In *ACM MM*, 2021.

[He *et al.*, 2018] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42:386–397, 2018.

[Huang *et al.*, 2018] Kun Huang, Yifan Wang, Zihan Zhou, Tianjiao Ding, Shenghua Gao, and Yi Ma. Learning to parse wireframes in images of man-made environments. In *CVPR*, 2018.

[Lin *et al.*, 2017] Tsung Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[Liu *et al.*, 2021] Hengyue Liu, Ning Yan, Masood Mortazavi, and Bir Bhanu. Fully convolutional scene graph generation. In *CVPR*, 2021.

[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-IJCNLP*, 2021.

[Neven *et al.*, 2018] Davy Neven, Bert De Brabandere, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Towards end-to-end lane detection: An instance segmentation approach. *Proceedings of IEEE Intelligent Vehicles Symposium*, pages 286–291, 2018.

[Neven *et al.*, 2019] Davy Neven, Bert De Brabandere, Marc Proesmans, and Luc Van Gool. Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth. In *CVPR*, 2019.

[Pratt, 2007] William K Pratt. *Digital Image Processing, Fourth Edition*. Wiley, 2007.

[Sachan *et al.*, 2017] Mrinmaya Sachan, Avinava Dubey, and Eric P. Xing. From textbooks to knowledge: A case study in harvesting axiomatic knowledge from textbooks to solve geometry problems. In *EMNLP*, 2017.

[Sachan *et al.*, 2020] Mrinmaya Sachan, Avinava Dubey, Eduard H. Hovy, Tom M. Mitchell, Dan Roth, and Eric P. Xing. Discourse in multimedia: A case study in extracting geometry knowledge from textbooks. *Computational Linguistics*, 45:627–665, 2020.

[Sandler *et al.*, 2018] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang Chieh Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018.

[Seo *et al.*, 2014] Min Joon Seo, Hannaneh Hajishirzi, Ali Farhadi, and Oren Etzioni. Diagram understanding in geometry questions. In *AAAI*, 2014.

[Seo *et al.*, 2015] Minjoon Seo, Hannaneh Hajishirzi, Ali Farhadi, Oren Etzioni, and Clint Malcolm. Solving geometry problems: Combining text and diagram interpretation. In *EMNLP*, 2015.

[Tian *et al.*, 2020] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: A simple and strong anchor-free object detector. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8828:1–1, 2020.

[Veličković *et al.*, 2018] Petar Veličković, Arantxa Casanova, Pietro Liò, Guillem Cucurull, Adriana Romero, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.

[Xu *et al.*, 2017] Danfei Xu, Yuke Zhu, Christopher B. Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *CVPR*, 2017.

[Ye *et al.*, 2020] Jun-Yu Ye, Yan-Ming Zhang, Qing Yang, and Cheng-Lin Liu. Contextual stroke classification in online handwritten documents with edge graph attention networks. *SN Computer Science*, 1(3), 2020.

[Ying *et al.*, 2021] Hui Ying, Zhaojin Huang, Shu Liu, Tianjia Shao, and Kun Zhou. Embedmask: Embedding coupling for one-stage instance segmentation. In *IJCAI*, 2021.

[Zhou *et al.*, 2019] Yichao Zhou, Haozhi Qi, and Yi Ma. End-to-end wireframe parsing. In *ICCV*, 2019.