symmetry

MDPI

Article

Solving Geometry Problems: A Text–Formula–Image Multimodal Parsing and Fusion Model

Pengpeng Jian [1,*], Zongxiang Song [1], Ting Song [2,*] and Yanli Wang [3]

[1] Information Engineering Institute, North China University of Water Resources and Electric Power, Zhengzhou 450046, China; x20231090803@stu.ncwu.edu.cn
[2] School of Information Technology, Henan University of Chinese Medicine, Zhengzhou 450046, China
[3] College of Marxism, Henan University of Economics and Law, Zhengzhou 450046, China; 20170060@huel.edu.cn
* Correspondence: jianpengpeng@ncwu.edu.cn (P.J.); songt@hactcm.edu.cn (T.S.); Tel.: +86-15838088763 (T.S.)

Abstract

Solving geometry problems is a critical challenge in education, for it demands the integration of textual semantic descriptions, mathematical formula logic and spatial graphical information, as well as rigorous geometric theorem application and stepwise logical deduction. These are core capabilities that underpin the realization of personalized intelligent tutoring and efficient educational resource allocation. Traditional geometry problem solving methods often suffer from deficiencies in accuracy and the fusion of text, formula and image features. Hence, this paper proposes a method of solving geometry problems based on a text–formula–image (TFI) multimodal parsing and fusion model. The TFI parser employs a self-attention multilayer Transformer to enhance the extraction of logical relations among geometric text expressions. Meanwhile, it parses formulas into tree structures to overcome the loss of formula structural features, which utilizes symbolic embedding and tree-structured encoding to preserve hierarchical logical information and yields unified formula representations via a multi-granularity fusion module. The TFI parser also leverages a Feature Pyramid Network (FPN) for the accurate detection of geometric and non-geometric instances, resolves the issues of blurred segmentation for slender geometric elements and the inaccurate localization of small-sized symbols through mask averaging and RoIAlign, and generates high-dimensional image features using DenseNet-121. The TFI multimodal fusion model integrates a contrastive learning mechanism and constructs fused feature representations by stacking self-attention and cross-attention layers. This design effectively narrows the semantic gap between text, formula, and image features, addressing the inadequacy of traditional fusion approaches in deep cross-modal feature alignment. An attention-augmented Gated Recurrent Unit (GRU) network processes the fused TFI features to produce target operation trees and geometry solutions, ensuring interpretable and precise reasoning performance. The proposed method is evaluated on the PGDP5K and GeoEval datasets, and it achieves an average accuracy of 59.63% in geometry problem solving, which validates its effectiveness. This paradigm offers a viable technical approach for uniformly modeling complex educational tasks, including geometry problem solving and timetable scheduling.

Keywords: geometry problem; automatic solver; text–formula–image parser; multimodal fusion

Check for updates

Academic Editor: Shuang Xu

Received: 5 January 2026
Revised: 13 February 2026
Accepted: 21 February 2026
Published: 10 May 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Intelligent education functions as a pivotal application of educational digitalization and holds a crucial position in the fields of artificial intelligence (AI) and education. Within its diverse research avenues, machine-based geometry problem solving has emerged as a prominent research hotspot in the realm of intelligent education, exerting a pivotal driving influence on advancements in this field. Geometry, as a core subject in primary and secondary school mathematics education, plays an irreplaceable role in cultivating students' spatial imagination, logical reasoning, and abstract thinking abilities. The intelligent solution of geometry problems serves as a vital component of personalized tutoring, which provides targeted guidance for students at different learning levels and conducts automatic analysis of error-prone problem types. It also acts as a key support for elevating teachers' teaching efficiency, which enables the automatic correction of geometric homework and the construction of intelligent question banks for teaching practice. Recently, propelled by the rapid evolution of deep learning and its associated technologies, substantial progress has been attained in research on geometry problem solving, thereby profoundly reshaping the learning paradigms of students and the teaching practices of teachers.

Existing research [1,2] on geometry problem solving has mainly focused on text and image parsing, as well as cross-modal fusion between the two, and has achieved initial results in machine understanding of such problems. However, obvious limitations remain: traditional image parsing methods are difficult to accurately identify complex geometric structures; mathematical formulas, a key information carrier for problem solving, are often ignored in multimodal fusion, leading to inadequate problem understanding. To mitigate these limitations, this study conducts targeted innovative research, including developing specialized parsers for complex geometric images and mathematical formulas, and ultimately establishing a text–formula–image (TFI) multimodal parsing and fusion framework to improve problem-solving accuracy.

In geometry problem-solving scenarios, problem descriptions typically consist of two components: textual narratives and mathematical formulas. Pretrained language models typified by BERT [3], have demonstrated considerable potential in tackling mathematical problems involving textual semantics. However, these models exhibit notable limitations in capturing the structured semantics inherent to mathematical formulas. This deficiency stems from their core design, which is primarily optimized for natural language sequence processing [4], making them ill-suited for handling the discrete symbol systems and inherent mathematical operational logics [5]. Consequently, such models fail to accurately convey critical information indispensable for reasoning, directly impairing the precise generation of geometric reasoning paths. Therefore, relying solely on text parsing techniques to extract formula features is insufficient in geometry problem-solving scenarios; the rapid advancement of deep learning technology has provided key technical support for developing a specialized formula parser to address this gap. Specifically, scholars have proposed deep-learning-driven methods for supporting our understanding of formulas [6]. Most of these methods initially adopted attention mechanisms but were constrained by single-scale limitations; they fail to fully utilize multi-granularity visual information, which is critical for understanding mathematically symbols of varying scales. This thus highlights the necessity of developing a more advanced specialized formula parser that can dynamically adapt to and integrate visual information across all granularity levels.

In the field of image analysis, traditional methods have long been employed for the recognition of geometric elements, such as the Hough transform. They often suffer from low accuracy in complex scenarios, particularly when lines or arcs overlap. Currently, the primary approaches for image feature extraction encompass deep learning models, pretrained vision models, and multitask learning paradigms. Among these, ResNet [7],

RetinaNet [8], and related architectures have emerged as mainstream solutions. Existing deep learning models continue to confront non-trivial challenges with the increasing diversification of tasks and the growing complexity of images. Specifically, their performance remains unsatisfactory in addressing small object detection and segmentation, complex background interference, and inter-instance mutual interference.

Furthermore, geometry problem solving necessitates the integration of multimodal information, encompassing text, formulas, and images. To derive solutions for geometry problems, information from these modalities must be effectively fused, and logical reasoning must be applied to generate answers. Although several multimodal fusion techniques have been proposed [9–15], they still suffer from prominent "semantic gaps" within the specific context of geometry problems. Existing techniques frequently fail to capture the deep semantic correlations between text, formulas, and geometric images, which hinders the accurate comprehension of problems and the generation of automated solutions. Thus, there is an urgent need to develop more robust methods that map features from diverse modalities into a unified semantic space, thereby strengthening their inter-modal correlations.

To address the aforementioned issues, this paper proposes a method of solving geometry problems based on a TFI multimodal parsing and fusion model. Its key contributions are summarized as follows:

- Owing to the structural discrepancies between natural language text and mathematical formulas, directly applying text parsing techniques for formula feature extraction yields unsatisfactory results. To mitigate this limitation, a dedicated formula parser is developed, which is specifically optimized for the unique structural characteristics of mathematical formulas. This design effectively enhances the performance of formula feature extraction and improves parsing accuracy.
- To resolve the low recognition accuracy of complex geometric images, the proposed geometry image parser framework performs detection and segmentation on individual instances within geometric images. This approach effectively overcomes the recognition bottlenecks of traditional methods in complex geometric scenarios, followed by targeted feature extraction from the segmented instances.
- In terms of multimodal fusion, a text–formula–image (TFI) multimodal fusion model based on geometric semantic constraints is proposed. Specifically, text-rewriting technology is leveraged to generate high-quality samples, which optimizes the alignment among text, formulas, and images while enhancing semantic consistency and feature expressiveness. Subsequently, hierarchical self-attention networks and cross-modal cross-attention mechanisms are employed to refine feature alignment, reduce redundant information, and highlight key features. Ultimately, this enables the precise fusion of TFI modalities, yielding high-quality multimodal fusion vectors.

The remainder of this paper is structured as follows: Section 2 reviews the related work in geometric text/formula parsing, geometric image parsing, and geometric multimodal fusion. Section 3 details the proposed methodology, including the overall research framework, TFI parser model, TFI multimodal fusion model, and geometry problem solver model. Section 4 presents the experimental setup, datasets, and comprehensive analysis of experimental results. Finally, Section 5 summarizes the main conclusions of this study and discusses future research directions.

2. Related Work

Effectively solving geometry problems requires machines to understand and reason over multimodal information, including textual descriptions, mathematical formulas, and geometric images. Significant research efforts have been devoted to parsing these individual

modalities and fusing them for joint reasoning. This section reviews the state-of-the-art in three key areas pertinent to our work: (1) geometric text and formula parsing, (2) geometric image parsing, and (3) geometric multimodal fusion. We critically analyze the strengths and limitations of existing approaches, highlighting the persistent challenges that motivate our proposed method.

2.1. Geometric Text and Formula Parsing

The geometry problem text typically consists of two parts: textual descriptions and mathematical formulas. Pretrained language models represented by BERT have performed well in traditional tasks such as text classification and dialogue generation. Their powerful language modeling capabilities have also prompted researchers to apply them to the solution of mathematical problems. Li et al. [16] achieved semantic clustering of equation prototypes through the semantic encoding of BERT; using MWP-BERT, the authors of [17] enhanced the model's numerical reasoning ability by strengthening arithmetic pretraining tasks. However, both methods rely on general text parsing frameworks and fail to address the structural particularities of mathematical formulas—an issue that our proposed dedicated formula parser explicitly overcomes by integrating symbolic embedding and tree-structured encoding, which is more targeted for geometric formula parsing than these general language models. Jian et al. [18] achieved the generation of mathematical word problem solution results by using kernel mean dynamic aggregation and dual decoder collaborative fusion technology. Despite these incremental advancements in handling textual semantics, existing pretrained language models exhibit notable deficiencies in extracting the structured semantics of mathematical formulas embedded in problem texts. Fundamentally, traditional text parsing models are optimized primarily for continuous natural language sequences, yet lack robust parsing for the discrete symbol systems and inherent operational logic of mathematical formulas. In geometry problems, formula symbols, algebraic expressions, and spatial relations together form a specialized grammatical structure involving mathematical symbols, operators, and variables. Interpreting these requires models to not only recognize individual symbol semantics but also grasp inter-symbol combinatorial logic. However, current pretrained language models are often limited to character-level parsing, failing to detect critical details like operator precedence and inter-symbol mathematical constraints. This leads that structured information is lost, directly undermining the generation of mathematically valid solution paths in subsequent reasoning. Thus, existing methods that directly use text parser for formula feature extraction perform poorly. Due to inherent structural differences between text and formulas, text parsing frameworks fail to capture formulas' unique structural traits. There is therefore an urgent need for a dedicated formula parser to boost the accuracy and effectiveness of formula feature extraction, and our TFI parser fills this gap by customizing the design for geometric formula structures rather than repurposing general text parsing architectures in existing works.

2.2. Geometric Image Parsing

Geometric images convey critical visual–spatial information that complements textual problem descriptions, making the recognition of basic geometric elements and the analysis of their spatial relations a fundamental prerequisite for geometry problem solving. Traditional recognition methods, particularly those based on Hough Transform, have long been prevalent for this task. Nevertheless, these traditional methods exhibit limited recognition accuracy when processing complex geometric images, particularly in cases involving overlapping lines or arcs. To address this limitation, image segmentation techniques can be introduced to enhance the performance of Hough-Transform-based methods in complex scenarios. The primary objective of image segmentation is to partition an image into se-

mantically meaningful regions or objects; this involves not only distinguishing between different object categories but also separating individual instances within the same category. Notable advancements in image segmentation include the Mask R-CNN framework proposed by He et al. [19], which builds on Faster R-CNN by incorporating a parallel branch for predicting object segmentation masks, thereby achieving efficient object detection and high-quality instance segmentation. In contrast, our geometric image parser is specifically optimized for such complex geometric scenarios, integrating mask averaging and RoIAlign to mitigate the performance degradation observed in Mask R-CNN, achieving more accurate segmentation of slender and overlapping geometric elements. Additionally, Ying et al. [20] develop a method that integrates the strengths of segmentation-based and proposal-based approaches: this single-stage instance segmentation method generates pixel and proposal embedding via an embedding module, then produces high-resolution masks through pixel-level clustering to mitigate detail loss. Despite this progress, most existing segmentation methods are optimized primarily for natural scene instance segmentation, yielding suboptimal performance in geometric image segmentation. Consequently, current approaches remain challenged in handling geometric image instance segmentation, especially in complex scenarios characterized by multiple overlapping lines or arcs. While this method improves small object segmentation in natural scenes, it lacks optimization for geometric-specific elements such as slender lines and overlapping arcs, resulting in suboptimal performance in geometry problem scenarios and this is an issue our approach directly addresses.

2.3. Geometric Multimodal Fusion

Geometry problems are inherently multimodal, requiring the integration of textual, formulaic, and visual information to enable effective reasoning and solution generation. The core of geometric problems lies in grasping the topological relationships and connectivity between basic elements such as points, lines, and circles, and multimodal fusion is an effective approach to boost the model's ability to capture such geometric structural characteristics. To accurately identify the topological and connectivity relations of points, lines, and circles, a single modality is far from sufficient, while multimodal fusion realizes the precise characterization of geometric structures by integrating complementary information from different modalities: the visual modality directly captures the intuitive morphological features and spatial positional relations of geometric elements such as points, lines, and circles, and the textual and formula modalities provide explicit logical constraints and formal descriptions for the topological dependencies between these elements. This cross-modal complementarity helps the model avoid one-sided judgment caused by single-modality limitations, accurately identify the intrinsic connections and topological dependencies between geometric elements, and thus fundamentally improve the accuracy of geometric structure understanding and subsequent problem solving. The effectiveness of multimodal fusion for enhancing the recognition of structural connectivity and topological relationships between spatial elements has been verified in related research fields. For example, Yuan et al. [21] proposed a multimodal attention fusion framework for remote sensing imagery, which adopts a dual-stream encoder architecture and a dedicated multimodal attention fusion module to capture fine-grained and cross-scale feature dependencies between spatial elements, effectively restoring the structural continuity and completeness of road networks and improving the recognition of connectivity relationships between spatial elements. Its core idea of using multimodal attention fusion to enhance the perception of spatial element topological to better capture the topological relationships and connectivity between geometric basic elements such as points, lines, and circles. Existing multimodal fusion strategies primarily fall into two categories: embedding image and text features

into a shared semantic space; processing individual modalities independently prior to cross-modal fusion. For instance, ALIGN [9] adopts a dual-encoder architecture to align visual and linguistic features in a shared space, yielding a transferable model via contrastive learning. While effective for general image–text matching, ALIGN ignores formula modalities, leading to poor performance in geometry problems that rely heavily on algebraic derivation and our TFI framework supplements this by incorporating formulas as a core modality. Jian et al. [22] solved geometric problems by feature learning and contrastive learning of multimodal data. Their fusion approach fails to capture formula semantics, a limitation that our dedicated formula parsing branch overcomes. Image–text contrastive learning [10] seeks to learn optimal representations of images and texts in the feature space, enabling the model to capture inter-modal commonalities and differences while matching image and text features within this space. Distinct from contrastive learning paradigms, Zhang et al. [11] proposed a hierarchical fusion mechanism of textual clauses image geometric primitives, which achieves cross-modal hierarchical alignment through structural semantic pretraining and addresses the semantic gap between geometric images and texts. Our TFI framework extends their hierarchical alignment idea to include formula clauses, establishing a three-modal text–formula–image hierarchical fusion. Despite substantial advancements across diverse fields, multimodal fusion of image and text features for geometry problem solving remains challenging. Specifically, the inherent semantic gap between textual and visual modalities in geometric contexts results in weak inter-modal correlation, which impairs multimodal data understanding and hinders machines from accurately interpreting problems and generating automated solutions. Thus, developing effective methods to map textual and visual feature representations into a unified semantic space is imperative, thereby enhancing their correlation. Recently, researchers have explored image–text feature fusion to improve geometry problem understanding and reasoning. VisualBERT [23] aligns text with image regions via a self-attention mechanism, enabling language–image region matching without explicit supervision signals while capturing syntactic relations. Its generic self-attention mechanism fails to resolve referential ambiguities in complex geometric images, a problem that our geometric domain-knowledge-guided attention alignment addresses. Inter-GPS [12] extracts formal language sets and symbolic reasoning to support the derivation of final solutions. Our TFI parser avoids rule-based limitations by using data-driven formula parsing and fusion, making it more robust to diverse geometric problem types than Inter-GPS. PGDP [13] introduces an improved instance segmentation method for geometric primitive extraction to enable problem solving. Our multimodal fusion model explicitly models text–formula–image interactions, bridging the gap between visual primitive extraction and formula-based reasoning that PGDP fails to address. Jian et al. [24] employed an improved RetinaNet and Graph Convolutional Network to achieve interpretable geometry problem solving. Additionally, they [25] accomplished the parsing and reasoning of geometric diagrams based on deep semantic fusion technology. Our work complements these two studies by adding formula as a core modality, enabling more comprehensive semantic fusion than their text–image-only or diagram-focused frameworks. GeoDRL [26] parses problem images and texts into a geometric logic graph before utilizing deep neural networks for decision making and Q-value estimation, alongside a symbolic system for logical inference. While these geometry tailored methods represent significant advances, they often rely on self-attention-based alignment mechanisms and still struggle to capture deep semantic relations across text, formulas, and geometric images. Three key limitations persist: firstly, they fail to fully resolve referential ambiguities between modalities; secondly, they lack explicit mechanisms to leverage geometric domain knowledge for constraining and guiding the fusion process; thirdly, fine-grained spatial or structural details are often lost during fusion. Consequently,

constructing robust methods that map multimodal features into a unified semantic space while leveraging geometric semantics to enhance inter-modal correlation and bridge the modality gap remains a critical research priority, one that our proposed TFI multimodal parsing and fusion model achieves by addressing all three limitations simultaneously and outperforming existing works through dedicated formula parsing, geometry-specific image segmentation and three-modal semantic fusion.

Although significant progress has been made in the research of machines solving geometry problems at present, there are still challenges in formula parsing, feature extraction for complex geometric images, and achieving deep and semantically consistent alignment and fusion of text, formula and image features. This study further addresses these unsolved challenges based on the above analysis, and the subsequent sections will detail the specific design and implementation of the TFI multimodal parsing and fusion model.

3. Methodology

3.1. Overall Research Framework

This paper aims to enhance the accuracy of feature extraction, fully integrate the features of text, formulas, and images, and thereby improve the accuracy of solving geometry problems. Hence, this paper proposes a method for solving geometry problems based on a TFI multimodal parsing and fusion model, and the overall framework is shown in Figure 1. The method mainly consists of three parts: the TFI parser model, the TFI multimodal fusion model, and the geometry problem solver model. The TFI parser extracts the features of text, formulas and images. The TFI multimodal fusion model fuses the three modal features. The solver based on Gate Recurrent Unit (GRU) provides the final solution to the geometry problem.

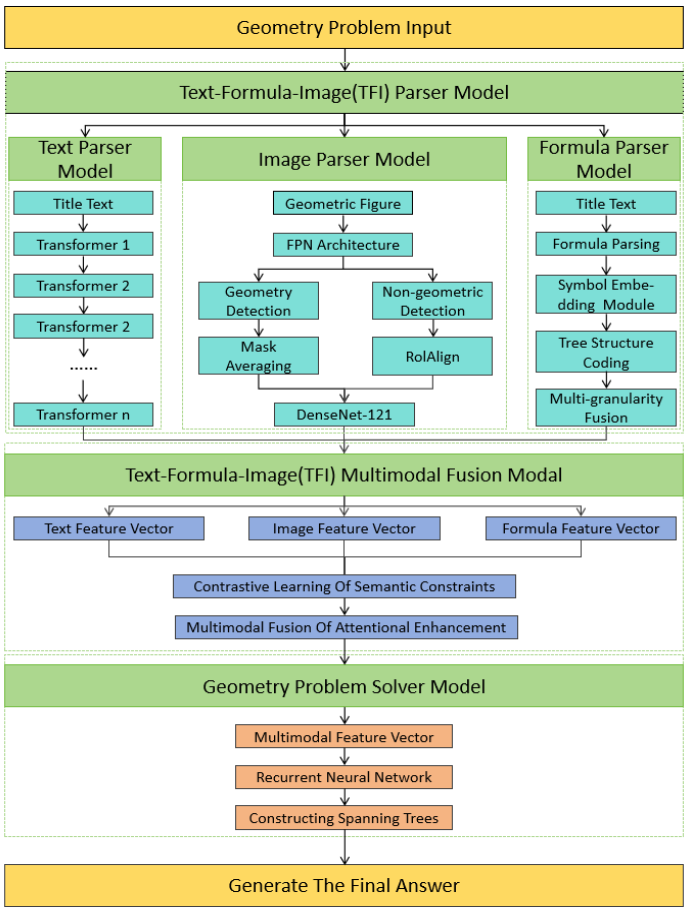


Figure 1. Overall framework.

In TFI parser model, a text parser that is based on the BERT architecture, is constructed using multilayer Transformers and self-attention mechanisms to learn the context features of the text. The accuracy of the context features is improved through alternating training of bidirectional prediction and sequence generation. The formula parser converts formulas into tree structures through formula parsing, encodes them using symbol embedding and tree structure encoding, and finally outputs a unified formula representation through a multi-granularity fusion module. For geometric images, the image parser regards the extraction of geometric instances as an instance segmentation problem and adopts the classic Feature Pyramid Network (FPN) [27] architecture as the backbone module for visual feature extraction. The P3–P7 layers of FPN are specifically used for the detection tasks of text and symbols, and a non-geometric detection module is introduced, which uses the anchor-free detection method Fully Convolutional One-Stage Object Detector (FCOS) [28] to improve the speed and efficiency of detection. In addition, the P2 layer is specifically used for the detection task of geometric shapes by the geometric segmentation module, which combines the semantic segmentation branch and the embedding segmentation branch to achieve instance segmentation of multiple types of geometric primitives. After the geometric instances and non-geometric instances are segmented, they are processed by mask averaging and RoIAlign methods, respectively, and then standardized to ensure that each feature has consistent distribution characteristics. Finally, they are input into DenseNet-121 [29] for feature extraction. Then, a fine-grained multimodal fusion model was designed, which utilizes multimodal contrastive learning to improve the semantic alignment accuracy among text, formulas, and images [30]. Subsequently, it introduces the self-attention mechanism and the cross-modal cross-attention mechanism, further enhancing the fine-grained alignment of features and optimizing the local semantic expression, to obtain the final high-quality multimodal features. The geometry solver generates the solution sequence step by step through the GRU network and ensures the legality of each operation symbol and formula symbol through the operation tree structure. The solver gradually builds the solution process by combining with the GRU hidden state and finally obtains the answer. During the training process, the negative log-likelihood loss function is used to optimize the generated sequence, and the Flooding method [31] is adopted to dynamically adjust the loss to prevent overfitting and ensure that the model is optimized at the best performance level.

3.2. TFI Parser Model

3.2.1. Text Feature Extraction

When dealing with the text input of geometry problems, the text extraction model regards the input as a text sequence $w = \{w_1, \cdots w_n\}$ similar to the BERT model in natural language processing. The model processes by converting position encoding, labels and task embedding into a sequence of text label embedding $W = \{E^t_{[CLS]}, E^t_{[w_1]} \cdots E^t_{[w_n]}, E^t_{[SEP]}\}$, where E and t represent the text embedding labels and the learning task embedding labels, respectively. Through multilayer Transformer and self-attention masking mechanism, the model can capture the contextual information in the text, thereby generating contextual tags that can accurately express the semantic of the text. Specifically, the BERT-based self-attention mechanism in this text parser processes geometry text via four core steps combined with multilayer Transformer: first, project each geometric text token embedding into query, key and value matrices for similarity calculation and attention weight generation via Softmax; second, apply bidirectional masking to restrict the model to attend only to valid bidirectional context and filter invalid tokens; third, the single-layer attention output is iteratively enhanced by multilayer Transformer, which assigns higher weights to key geometric terms and suppresses redundant words; finally, fuse the attention-enhanced

features with task embedding vector w_t^{task}, to retain geometry text semantics while distinguishing downstream tasks. The learning task embedding vectors w_t^{task} are treated as part of the input of the BERT model together with the sequence of text labels embedding vectors. Here, implicit features are defined as latent geometric and semantic correlations that cannot be directly obtained from raw text; they are extracted by the multilayer Transformer and self-attention mechanism followed by BERT fusion and task embedding separation. In subsequent tasks, the learned task embedding vectors can be separated from the implicit features $h^t = BERT(w, w_t^{task})$ of the text output to mark and handle different downstream tasks, while sharing the implicit features of the text. The text parser model is shown in Figure 2.

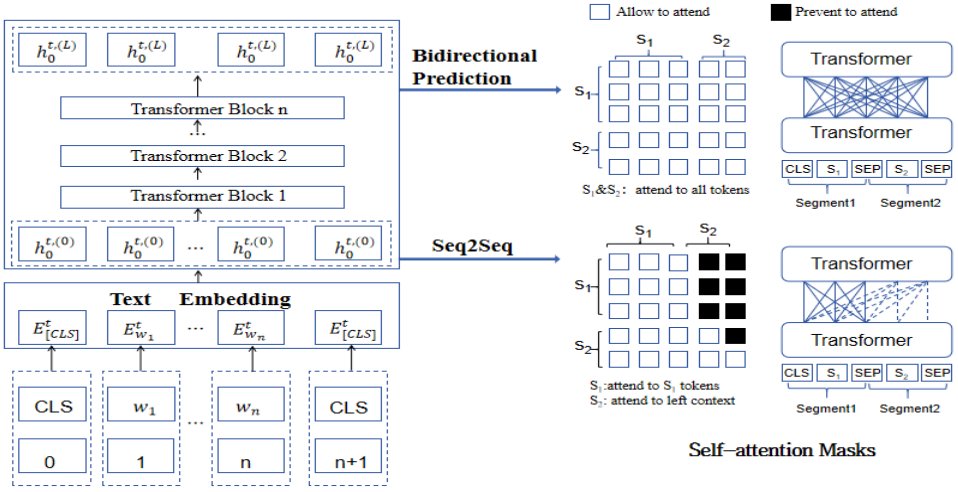


Figure 2. Text parser model.

In Figure 2, dashed frames mark boundaries of core functional modules. Solid arrows indicate the data flow direction of the model. For the self-attention module, solid lines denote fully bidirectional self-attention adopted for the bidirectional prediction task, enabling full contextual interaction of all tokens. Dashed lines denote unidirectional causal self-attention designed for the sequence-to-sequence auto-regressive generation task, restricting target tokens to attend only to left context and previous tokens. In the attention mask matrix, white squares allow attention calculation, while black squares mask invalid positions to prevent attention calculation.

In order to better learn the context-dependent token representations of text features, the model will adopt bidirectional prediction (BP) [32] and sequence-to-sequence (Seq2Seq) [33] tasks for training. The model employs a specific self-attention masking mechanism to control the context content. To enhance the language learning process, the model also utilizes the syntactic–semantic models [34,35] from the previous research to identify semantic complete phrases in the question text. During the training process, the model always samples complete word or phrase sequences rather than individual token markers. This process is alternately trained between the BP and Seq2Seq generation objectives to ensure that the model can balance learning between the two, thereby obtaining the context-dependent token representations of text features.

For the BP task, iterative span sampling is conducted on the sequence of text token embedding until approximately 15% of the tokens are selected. This 15% is an empirically optimized value for geometry text processing; it can be adaptively adjusted according to the scale of the geometry text dataset and the complexity of geometric semantic expressions, as a too low ratio fails to mine sufficient contextual information while too high a ratio would destroy the integrity of key geometric semantic information, and 15% achieves a

balance between the two. By randomly choosing the span of characters from the geometric distribution l-Geo(p), Among $l_{max} = 10,\ p = 0.2$, the markers in the selected area will be replaced by masked markers, random markers or original markers with probabilities of 80%, 10% and 10%, respectively. The objective is to predict the masked tokens w_m based on the context $w_{\backslash m}$ and train the model by minimizing the negative log-likelihood loss:

$$L_{BP}(\theta) = -E_{w \in D} log P_\theta(w_m | w_{\backslash m}) \quad (1)$$

Here, θ stands for the trainable parameters. Each word or phrase w is sampled from the training set D. w_{m} is the original token in the masked region. $w_{\backslash \mathrm{m}}$ is the contextual information of the text. For the BP task in geometry text processing, other potential loss functions have obvious limitations: the weighted version is unnecessary for basic contextual learning, cross-entropy loss is essentially equivalent with no extra benefits, and contrastive loss brings high computational complexity. This negative log-likelihood loss is thus selected for its core advantages: it is well-suited for masked token prediction, features stable gradient propagation and low computation, and is compatible with BERT's MLM to ensure training convergence efficiency.

The Seq2Seq task iteratively samples segments from the text token embedding sequence $\{E^t_{[CLS]}, E^t_{[w_1]} \cdots E^t_{[w_n]}, E^t_{[SEP]}\}$ until 25% of the budget is consumed. In each iteration, the model selects a segment length from the uniform distribution l-U(4, 32), and then samples segments based on this length [33]. Each selected segment $\{E^t_{[w_i]} \cdots E^t_{[w_j]}\}$ will be augmented with special start and end markers [CLS] and [SEP], thereby forming a new sequence $\{E^t_{[CLS]}, E^t_{[w_i]} \cdots E^t_{[w_j]}, E^t_{[SEP]}\}$. The selected segments will be deleted from the original text, and the resulting segments form the target sequence T, while the remaining parts are concatenated to form the source sequence S. Subsequently, the model will be trained on the source sequence to generate the target sequence under the condition of auto-regressive:

$$L_{Seq2Seq}(\theta) = -E_{(S,T) \in D} log P_\theta(T|S) \quad (2)$$

Here, θ is a trainable parameter. The target sequence and the source sequence (S, T) in the text are both sampled from the training set D. The Algorithm 1 for text feature extraction is as follows [32,33]:

Algorithm 1: Text feature extraction algorithm.

Input: Text sequence $w = \{w_1, \cdots w_n\}$ and learning task embedding vector w_t^{task}
Output: Text feature vector

1. Add [CLS] to the front of the sequence $w = \{[\mathrm{CLS}], w_1, \cdots w_n\}$
2. Add [SEP] to the end of the sequence $w = \{[\mathrm{CLS}], w_1, \cdots w_n, [SEP]\}$
3. Convert position encoding, labels, and task embedding into text label embedding sequence $W = \left\{E^t_{[CLS]}, E^t_{[w_1]} \ldots E^t_{[w_n]}, E^t_{[SEP]}\right\}$
4. Input W into the multilayer Transformer encoder to obtain $h^t = \left\{h_0^{t,(L)}, h_1^{t,(L)}, \cdots h_n^{t,(L)}, h_{n+1}^{t,(L)}\right\}$
5. Return text feature vector

3.2.2. Formula Feature Extraction

Aiming to capture the structural complexity and symbolic semantic diversity of mathematical formulas, this paper proposes a hierarchical formula feature extraction framework. The framework consists of a formula parsing module, a symbol embedding module, a tree structure encoding module and a multi-granularity fusion module. Through the collab-

orative work of each module, the framework can effectively solve the limitations of the traditional sequence model in formula structure modeling, multiscale symbol distribution and semantic association reasoning tasks, showing strong advantages. The formula parser model is shown in Figure 3.

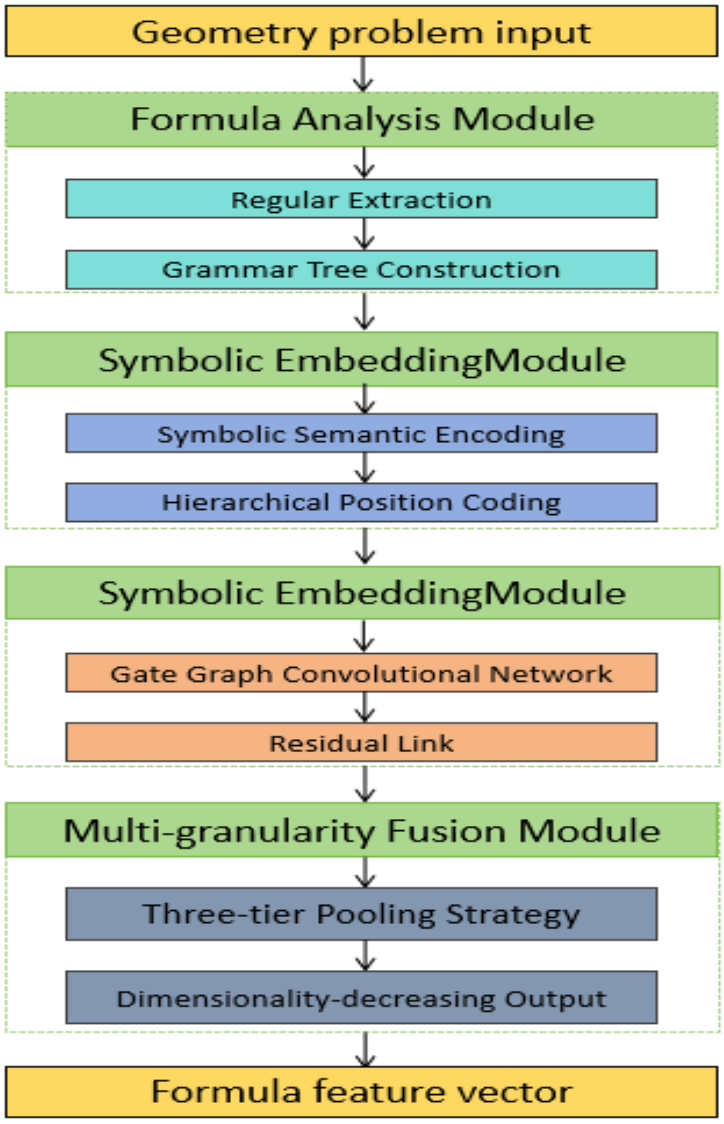


Figure 3. Formula parser model.

In the process of formula localization and standardization, the primary step is to achieve accurate identification and extraction of various formula structures, including typical mathematical expression forms such as equations, inequalities, and fractions, by relying on regular expression technology. By establishing a targeted regular expression rule system, this process enables the system to effectively match mathematical expressions in different representations. Among these procedures, the explicitization of implicit multiplication requires a customized pattern matching strategy to accurately identify scenarios where numbers are adjacent to geometric symbols or letters, such as converting "3AB" to "3*AB". This clarifies the multiplicative relation between coefficients and variables, ensuring that the expression format meets the parsing requirements of SymPy. On this basis, an improved SymPy parser is adopted to conduct lexical and syntactic analysis. As a built-in tool of the open-source Python 3.10 symbolic math library SymPy, it parses standardized mathematical formula strings into machine-understandable symbolic struc-

tures. We use it for its native support for geometric formula analysis, high customizability for geometric symbol priority marking, and seamless generation of abstract syntax trees to enable subsequent geometric semantic integration. In the lexical analysis phase, the standardized formula string is split into several syntax units, and geometric symbols are marked as high-priority through a custom operator priority dictionary. In the syntactic analysis phase, the sequence of syntax units with priority labels is used as input, and the parsing of the syntactic structure is completed in accordance with mathematical operation logic and custom priority rules, ultimately generating an abstract syntax tree. A depth-first traversal is performed on the abstract syntax tree generated by syntactic analysis and type labels are added to all geometric symbol nodes, thereby constructing a composite syntax tree structure that integrates both mathematical operation logic and geometric semantic information.

Then the symbolic embedding module is used to map each node in the expression tree to symbolic semantic vector and structural position coding [36,37], so as to capture the deep meaning and tree structure of mathematical expressions. In this module, this work designs a structure-aware hybrid coder. According to the semantic complexity and internal structure characteristics of mathematical symbols, the optimal coding method is dynamically selected: atomic symbol coding, for atomic symbols without internal structure such as operators and single-letter variables, shared embedding table is adopted for encoding. This formula maps atomic symbols without internal structure to a d-dimensional semantic vector space, with the encoding defined as:

$$h_v = E_{\text{base}}(v) \in \mathbb{R}^d \tag{3}$$

Here, h_v denotes the semantic vector of the atomic symbol v and E_{base} represents the shared embedding matrix, which enables efficient and consistent encoding of atomic symbols in geometric formulas with reduced model parameters.

For the geometric structure encoding, regarding the geometric image declarations and compound angles, such as $\triangle$ABC, $\angle$BCD, a bidirectional GRU model is used to encode their vertex sequences. This formula encodes the vertex sequences of geometric structures by BiGRU and superimposes type embedding to capture geometric semantic information, defined as:

$$h_{shape} = BiGRU(E_{base}(v_1), \ldots, E_{base}(v_k)) + E_{type}(s) \tag{4}$$

In this formula, h_{shape} is the semantic vector of geometric structures and $E_{\text{type}}(s)$ is the type embedding matrix, which captures the sequential semantics of geometric vertices and enhances the representation of geometric attributes effectively.

For combination symbols with specific semantics, such as line segment AB with a unit constant of 5 cm, the semantic combination encoding retains their overall semantics through the projection network. This formula fuses symbol embedding and unit encoding via MLP to preserve the overall semantics of unit-related combination symbols, with the encoding formula as:

$$h_{combo} = MLP([E_{base}(x); E_{base}(y); \phi(u)]) \tag{5}$$

Here, h_{combo} is the semantic vector of combination symbols and $\phi(u)$ is the one-hot encoding of units, which fuses symbol and unit semantics to preserve the integral meaning of geometric combination symbols.

To further capture the structural information of the expression tree, the module designs hierarchical positional encoding. This method combines depth encoding and sibling encoding to explicitly model the topological relation of the tree structure. Depth encoding encodes based on the depth of the tree, starting from a depth of 0 at the root node, and increasing by 1 for each lower level, mapping hierarchical differences through a learnable

embedding matrix $PE_{\text{depth}} \in \mathbb{R}^{D\times d}$; sibling encoding assigns positions based on the preorder index of the child nodes under the same parent node, generating an embedding matrix $PE_{\text{sib}} \in \mathbb{R}^{P\times d}$. Ultimately, the positional encoding is a combination of these two. This formula integrates depth encoding and sibling encoding to model the topological relation of the expression tree, with the hierarchical positional encoding defined as:

$$PE_{(d,p)} = PE_{depth}(d) + PE_{sib}(p) \tag{6}$$

In this expression, $PE_{(d,p)}$ is the hierarchical positional encoding, with d as node depth and p as sibling node index, which explicitly models the topological structure of geometric formula expression trees.

The final node feature is the element-wise sum of the semantic encoding and the position encoding. This formula combines symbolic semantic encoding and hierarchical positional encoding to generate the final node feature, defined as:

$$h_{final} = h_{semantic} + PE(d,p) \tag{7}$$

Here, h_{final} is the final fused feature of tree nodes and $h_{semantic}$ is the symbol semantic encoding, which fuses semantic and structural information to enhance the model's understanding of complex geometric mathematical formulas. In this way, the model incorporates the hierarchical structure information of the tree into the semantic embedding of the symbols, thereby enhancing its understanding and processing capabilities of mathematical expressions.

Then, we use the tree structure coding module and gate graph convolutional network to encode the syntax tree efficiently [38,39]. The model expresses the different syntactic and geometric relations between nodes by defining three types of edge relations, namely, parent–child edge, sibling edge and geometrically associated edge. Parent–child edges are used to model the dominant relation between operators and operands, sibling edges capture interactions between sibling nodes, and geometrically associated edges encode constraints between geometric elements. The feature update of each node is carried out through the gating mechanism. The features of neighbor nodes are aggregated by weighted summation, and the fusion ratio of historical features and new features is dynamically controlled by Sigmoid function. In this process, the features of key geometric nodes are retained, such as angle and side length, and ordinary nodes are appropriately updated. The residual connection adopted in the following process effectively alleviates the problem of gradient disappearance in the deep graph network through cross-layer feature fusion. Finally, the node characteristics are obtained.

In order to capture the details and global semantics of the formula more comprehensively, the multi-granularity fusion module adopts the three-level pooling strategy to carry out feature fusion. Firstly, the mean pooling operation of leaf nodes preserves the details of the numerical changes in the formula by calculating the arithmetic mean of all leaf node features. Secondly, the subtree maximization operation maximizes the root node features of each subtree element by using elements to highlight key operational structures such as multiplication and addition nodes. Finally, attention weight pooling introduces learnable parameters to dynamically focus on important nodes by calculating the weight of each node. On the basis of these three pooling methods [40,41] the features are spliced and represented uniformly by dimensionality reduction. The Algorithm 2 for formula feature extraction is as follows [36–41]:

Algorithm 2: Hierarchical formula feature extraction algorithm.

Input: Original mathematical formulas
Output: Formula feature vector

1. Extract formula structures via regular expressions, convert implicit multiplication to explicit form.
2. Generate a syntax tree using a modified Sympy parser.
3. Generate symbolic semantic vectors: if atomic symbols: Encode via a shared embedding table h_v; else if geometric structures: Encode vertex sequences using BiGRU and superimpose type embedding get h_{shape}; else if unit-related combination symbols: Fuse symbol embedding and unit encoding via MLP get h_{combo}.
4. Calculate hierarchical positional encoding.
5. Generate the final node feature.
6. Define parent–child edges, sibling edges, and geometrically associated edges; aggregate neighbor features by weighted summation, update features via a gating mechanism, and introduce residual connections.
7. Mean pooling of leaf nodes, max pooling of subtree root nodes, and attention weight pooling; concatenate the pooled features, then standardize and reduce the dimension of the concatenated features.
8. Return the formula feature vector.

3.2.3. Image Feature Extraction

Regarding geometric image feature extraction as an instance segmentation problem, considering that geometric primitives such as lines and arcs usually present elongated and overlapping features, the instance segmentation method based on bounding boxes is not applicable to this task. To address these issues, a geometric image parser framework is proposed, which draws on the core module design of PGDPNet [13] while tailoring it for the specific scenario of geometry problem images. It is specifically designed for processing geometric shapes and symbolic texts in images, and is particularly suitable for scenarios with complex geometric layouts and different scales. It is the synergetic operation of the framework's functional modules that enables the effective solution to the geometric instance segmentation problem, with detailed mechanisms elaborated in the following content. This framework combines multiple modules such as FPN, non-geometric detection module, geometric segmentation module, DenseNet-121, and through collaborative work, and it can efficiently extract geometric features from images. The geometric image parser model is shown in Figure 4.

In this framework, the backbone network of the image parser adopts the FPN architecture for extracting visual features. FPN gradually extracts different scale features from coarse to fine through multi-level feature pyramids, ensuring that various geometric shapes in the image can be effectively captured and recognized. The P3-P7 layers of FPN are specifically used for handling symbol and text detection tasks in images. By extracting features of different resolutions, it can identify text and symbol elements at different scales, adapting to variations in the size of various symbols and texts. Since symbols and texts are non-geometric instances, traditional geometric shape detection methods often have low efficiency when dealing with such tasks. To accurately identify these symbols and texts, the framework introduces a non-geometric detection module, using the anchor-free detection method FCOS, aiming to improve the efficiency and accuracy of detection. The non-geometric detection module consists of three key branches: the regression branch predicts the position of each pixel point of the target; the center branch calculates the distance

of each pixel from the target center to help improve detection accuracy; the classification branch classifies each pixel point and outputs the probability that it belongs to a certain symbol or text. Through the collaborative effect of these three branches, the non-geometric detection module can avoid the complexity of traditional anchor-based detection methods, thereby improving the speed and accuracy of symbol and text detection. During the training process, the loss function L_{FCOS} integrates the regression loss L_{reg}, the center loss L_{cns} and the classification loss L_{cls} to ensure the efficiency and accuracy of the model in symbol and text detection:

$$L_{FCOS} = L_{reg} + L_{cns} + L_{cls} \tag{8}$$

Here, L_{reg}, L_{cns} and L_{cls} denote regression, center and classification loss, respectively, and this formula fuses the three types of loss to ensure the efficiency and accuracy of non-geometric instance detection for symbolic texts in geometric images.

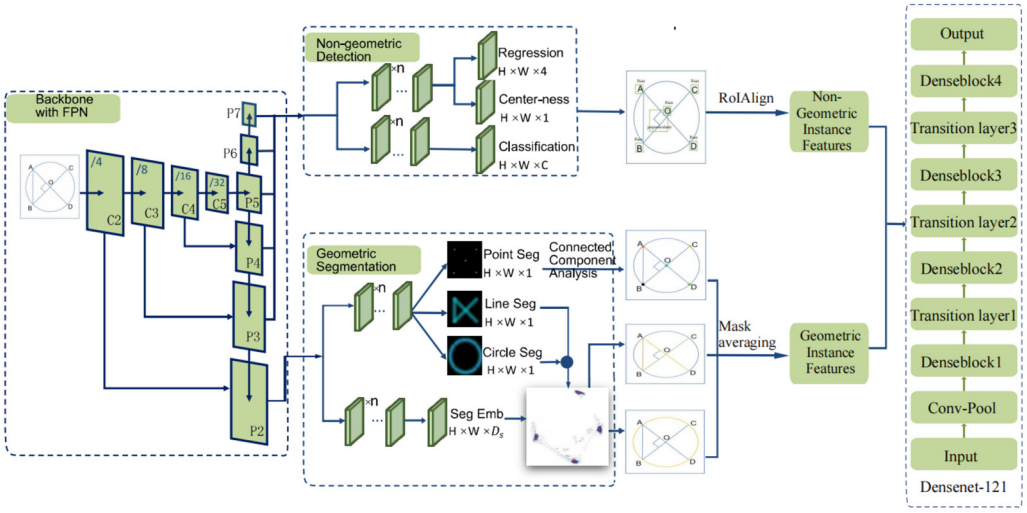


Figure 4. Image parser model.

At the P2 layer, the image parser employs a geometric segmentation module, which extracts structured features of geometric shapes and conducts in-depth analysis of their geometric attributes to achieve precise classification and understanding of the shapes. This module is fine-tuned for slender and overlapping geometric elements commonly seen in geometry problems, enhancing the segmentation robustness for lines, arcs and other primitives. Especially for complex geometric layouts and slender geometric shapes, traditional instance segmentation methods based on bounding boxes are not applicable. Therefore, the geometric segmentation module adopts an innovative instance segmentation method that combines two branches: semantic segmentation and embedding segmentation, enabling effective segmentation of multiple types of geometric primitives.

The semantic segmentation branch is responsible for binary segmentation of geometric shapes, using the weighted binary cross-entropy loss to optimize the segmentation effect. This formula defines the weighted binary cross-entropy loss for the semantic segmentation branch, which balances positive and negative pixel weights to improve segmentation accuracy of geometric primitives:

$$L_{bs*} = \frac{-w*}{Mmap} \sum_{i=1}^{Mmap} y_i^* log(p_i^*) + (1 - y_i^*) log(1 - p_i^*) \tag{9}$$

Here, $w*$ is the positive–negative pixel weight, $Mmap$ is the number of segmentation map pixels, y_i^* and p_i^* are the true label and predicted probability of pixels, and $*$ represents points, lines, and circles; this formula balances pixel weights to improve the binary segmentation accuracy of slender geometric primitives.

By using weighted loss, the model can more accurately segment geometric shapes such as straight lines, circles, and triangles. For each geometric shape category, there is a separate semantic segmentation loss. In the formula, * represents the three categories of points, lines, and circles, w is the weight ratio for balancing positive and negative class pixels, M_{map} is the number of pixels in the segmentation map. L_{bs} is the total sum of semantic segmentation losses for all categories:

$$L_{\text{bs}} = L_{\text{bsp}} + L_{\text{bsl}} + L_{\text{bsc}} \tag{10}$$

Here, L_{bsp}, L_{bsl} and L_{bsc} denote the semantic segmentation loss of points, lines and circles, respectively, and this formula integrates the loss of multiple geometric primitives to achieve refined semantic segmentation for different geometric shapes.

The embedding segmentation branch further distinguishes the geometric shapes of different instances by optimizing the embedding space between instances and by distinguishing them through instance-specific features. This method can effectively reduce instance confusion and thereby improve segmentation accuracy, especially when dealing with complex geometric layouts. The specific loss function is:

$$L_{dist} = \frac{1}{N_{lc}(N_{lc}-1)} \sum_{n_1=1}^{N_{lc}} \sum_{\substack{n_2=1 \\ n_2 \neq n_1}}^{N_{lc}} [2\delta_{\text{d}} - ||\mu_{n_1} - \mu_{n_2}||]_+^2 \tag{11}$$

$$L_{var} = \frac{1}{N_{lc}} \sum_{n=1}^{N_{lc}} \frac{1}{M_n} \sum_{i=1}^{M_n} [||\mu_n - x_i|| - \delta_v]_+^2 \tag{12}$$

L_{dist} is the discriminative loss between instances, N_{lc} is the total number of line and circle instances, δ_{d} is the center distance threshold, and μ is the instance embedding center. L_{var} is the variation loss of the embedding branch, M_n is the number of pixels in a single geometric instance, δ_v is the embedding radius threshold, and x_i is the pixel embedding vector; which is used to enhance the discrimination between instances and reduce confusion [42,43]. Among them, N_{lc} is the sum of the instance numbers of line N_l and circle N_c, M is the number of pixel instances, x is the number of pixel embedding, and μ is the center of instance embedding. By default, the center distance threshold δ_d is set to 1.5, and the embedding radius threshold δ_v is set to 0.5. Through the collaboration of these two branches, the geometric segmentation module can achieve efficient instance segmentation of multiple types of geometric primitives.

Finally, the geometric segmentation module combines these three loss functions to accurately segment and identify various geometric shape instances in complex geometric layouts. The total loss of the geometric segmentation module is:

$$L_{\text{ins}} = L_{\text{bs}} + L_{\text{dist}} + L_{\text{var}} \tag{13}$$

Here, L_{bs}, L_{dist} and L_{var} denote semantic segmentation, inter-instance discrimination and intra-instance variation loss, respectively; this formula fuses three losses to ensure effective instance segmentation of multiple geometric primitives in complex geometric layouts.

In order to further enhance the extraction and representation of different object features, the outputs generated by the geometric segmentation module and the non-geometric object segmentation module are refined. This process covers two main object categories: geometric instances and non-geometric instances. For geometric instances after segmentation processing, the features can be represented by the average of the masks. The mask averag-

ing operation generates a fixed-length feature vector by performing spatial averaging on the mask regions of the geometric objects. As shown in Formula (14):

$$VL_i^{geo} = B^T \frac{mask_i}{||mask_i||1} \tag{14}$$

Here, B is the visual feature map, $mask_i$ is the geometric instance mask, and $||\cdot||1$ is the L1 norm; this formula generates fixed-length geometric feature vectors via mask spatial averaging to retain structured features of geometric instances.

For non-geometric instances, the RoIAlign method is adopted for feature extraction. This method can precisely extract information from the regions of interest and ensure the consistency of features at different scales. As shown in Formula (15):

$$VL_i^{non-geo} = RoIAlign(B, box_i) \tag{15}$$

Here, box_i is the detection box of non-geometric instances, and RoIAlign is the region of interest alignment method; this formula accurately extracts regional features of non-geometric instances to ensure feature consistency across different scales.

The extracted geometric instance features and non-geometric instance features will undergo standardization processing, adjusting the mean of each feature vector to 0 and the standard deviation to 1, to eliminate the scale differences among different feature dimensions and ensure feature consistency. The standardized features are concatenated into joint features and adapted to the input of DenseNet-121 through fully connected layers and reshaping operations. Subsequently, they are input into DenseNet-121 for further processing. DenseNet-121 effectively combines low-level detail features with high-level global information, enhancing the model's expressive power and robustness. The final generated high-dimensional feature vectors contain rich geometric structures, spatial relations, and symbolic information, capable of providing fine and comprehensive feature representations for different tasks. The Algorithm 3 for image feature extraction is as follows [27–29]:

Algorithm 3: Image feature extraction algorithm.

Input: Original image
Output: Image feature vector

1. Use FPN as the backbone network to extract multiscale feature maps {P2, P3, . . ., P7}
2. Input P3–P7 to the non-geometric instance detection module FCOS to generate detection boxes and category labels for non-geometric instances.
3. Input P2 to the geometric instance detection module to generate instance embedding vectors and binary segmentation masks.
4. Perform mask averaging on geometric instances to generate geometric instance features VL_i^{geo}.
5. Use RoIAlign on non-geometric instances to generate non-geometric instance features $VL_i^{non-geo}$.
6. Standardize the geometric instance features and non-geometric instance features, then concatenate them into a joint feature, and adapt it to the input of DenseNet-121 through a fully connected layer and reshaping.
7. Further process with DenseNet-121 to generate high-dimensional features.
8. Return image feature vector.

3.3. TFI Multimodal Fusion Modal

In TFI multimodal fusion, two models are designed, namely matrix factorization bilinear pooling (MFB) fusion and fine-grained fusion, which differ in the granularity of feature interaction. MFB fusion mainly focuses on global feature interaction, performing dimensionality reduction through sum pooling and other methods, and adjusting features using gating mechanisms. However, it is insufficient in capturing local fine-grained matches. In contrast, fine-grained fusion achieves precise local feature interaction through attention mechanisms, effectively filtering out redundant information. Specifically, self-attention processes information within a single modality, while cross-attention dynamically associates local features across modalities, thereby better handling the complex correspondences between modalities. In the final experiment, the fine-grained fusion model will be used as the multimodal fusion method in this paper, and the MFB fusion will be used as the ablation experiment for analysis and comparison.

3.3.1. MFB Fusion

After the features of text, formulas, and images are extracted, since text and formulas usually have stronger semantic connections, especially in geometry problems where formulas are an important part of the text, fusing these two first may help the model better understand the semantic meaning of the problem, and then combine the geometric shape information in the image. This sequence may be more in line with human problem-solving logic, that is, to first understand the text and formulas in the problem and then analyze it in combination with the graphics. Based on this, a hierarchical fusion is proposed, which is divided into two stages: first, fuse the text and formulas, and then fuse the result with the image.

In the implementation process of hierarchical fusion, to eliminate the dimensional differences between modalities and enhance the feature expression ability, the text feature T and formula feature F are first linearly projected to map them to a unified latent space, obtaining T′ and F′. This formula realizes linear projection of text and formula features, eliminating dimensional differences and mapping them to a unified latent space for subsequent fusion:

$$\begin{aligned} T' &= W_T T + b_T, \quad W_T \in \mathbb{R}^{d_1 \times d_t}, \quad b_T \in \mathbb{R}^{d_1} \\ F' &= W_F F + b_F, \quad W_F \in \mathbb{R}^{d_1 \times d_f}, \quad b_F \in \mathbb{R}^{d_1} \end{aligned} \tag{16}$$

Here, d_1 represents the dimension after projection, W_T and W_F are learnable parameter matrices, and b_T and b_F are bias terms. This step ensures that the text and formula features interact within the same dimensional space, laying the foundation for subsequent bilinear pooling.

Next, the MFB model is utilized to further fuse these two modalities through bilinear mapping. The MFB model captures fine-grained cross-modal interactions through low-rank decomposition, enhancing the expressive power of the fused features. The bilinear weight tensor is decomposed into two low-rank matrices through low-rank decomposition: $U \in \mathbb{R}^{d_1 \times Ko_1}$, $V \in \mathbb{R}^{d_1 \times Ko_1}$. Here, U and V respectively correspond to the interaction basis vectors of the text and the formula. Element-wise interaction calculation: Perform a linear transformation on the projected features and calculate the Hadamard product. This formula calculates the element-wise interaction between projected text and formula features using the Hadamard product, capturing fine-grained cross-modal correlations:

$$M = \left(U^\top T'\right) \odot \left(V^\top F'\right) \in \mathbb{R}^{Ko_1} \tag{17}$$

Here, $\odot$ represents element-wise multiplication. The high-order correlations between the modal features are captured approximately through outer product.

Subsequently, sum pooling is applied to reshape the interaction matrix M into a $K \times o_1$ dimension. Summation is performed along the rank direction to compress the dimension. The calculation formula is:

$$z_{tf} = \text{SumPooling}(M, K) = \sum_{k=1}^{K} M_{k,:} \in \mathbb{R}^{o_1} \tag{18}$$

Here, SumPooling denotes sum pooling operation, and K is the pooling dimension; this formula compresses feature dimensions by summing along the rank direction while retaining cross-modal interaction information.

Power normalization and L2 normalization operations are introduced in the MFB module to prevent the model from getting stuck in local minima, thereby enhancing the model's stability and generalization ability. The corresponding formula definitions are as follows:

$$z_{tf} = sign(z_{tf})|z_{tf}|^{0.5} \tag{19}$$

$$z_{tf} = z_{tf}/||z_{tf}||_2 \tag{20}$$

Here, $sign(\cdot)$ denotes the sign function; this formula performs power normalization to prevent the model from falling into local minima and enhance training stability. $||\cdot||_2$ denotes the L2 norm; this formula applies L2 normalization to standardize feature scales and improve model generalization ability.

To further enhance the semantic complementarity between text and formulas, a gating mechanism is introduced to adaptively adjust the contribution ratio of the fused features and the original features. The fusion characteristics of the output of the first stage z_{tf}^{gated} is obtained.

The fused features z_{tf}^{gated} output from the first stage are projected a second time with the image features I into a unified space. Then, the MFB framework is reused to fuse the joint features of text and formula with the image features, ultimately obtaining the multimodal fusion features of text, formula, and image.

3.3.2. Fine-Grained Multimodal Fusion Model

To enhance alignment accuracy and reduce potential errors, the alignment process can be optimized through multimodal contrastive learning. The core objective of contrastive learning is to ensure that semantically related samples have a smaller distance in the shared embedding space, while semantically unrelated samples maintain a larger distance, thereby achieving more precise semantic alignment.

Firstly, the multimodal contrast model maps the feature representations of text, formulas, and images to a unified embedding space through fully connected layers. Secondly, due to the lack of explicit structured relations between formula symbols and visual elements in images, directly calculating the contrastive loss between formulas and images may amplify noise. Therefore, text serves as a stabilizing central modality in this process. By using text as a bridge, it can simultaneously associate formulas and images, filtering out some noise. Specifically, during training, positive sample pairs of text–formula and text–image are randomly selected, and the contrastive loss function is optimized. This approach prompts related examples to be closer in the embedding space, while the distance between unrelated examples is increased. To further improve alignment quality, text-rewriting techniques are employed to generate a large number of positive and negative examples from the original text–formula and text–image pairs, rather than simply randomly sampling negative samples. Positive samples are those semantically matching the given text–formula or text–image pairs, while negative samples are those that do not match. Through this method of

generating positive and negative samples, the problem of inaccurate alignment caused by random negative samples is avoided, thereby more effectively optimizing the contrastive loss function and enhancing the precision of semantic alignment. The contrastive loss functions for text–formula and text–image can be expressed as follows:

$$\begin{aligned} \mathcal{L}_{\text{t2f}} &= -\log \frac{e^{s(t,f^+)/\tau}}{e^{s(t,f^+)/\tau} + \sum_{f^-} e^{s(t,f^-)/\tau}} \\ \mathcal{L}_{\text{t2i}} &= -\log \frac{e^{s(t,i^+)/\tau}}{e^{s(t,i^+)/\tau} + \sum_{i^-} e^{s(t,i^-)/\tau}} \end{aligned} \tag{21}$$

Here, $s(\cdot)$ denotes similarity, τ is temperature coefficient, f^+/f^- and i^+/i^- are positive/negative samples; this formula optimizes text–formula and text–image semantic alignment.

The main objective of the second stage is to establish strong associations between text and formulas, as well as between text and images. During this stage, since there is a lack of direct supervisory signals between formulas and images, the formula–image contrast loss is not directly calculated. At this point, the model needs to be forced to learn the role of text as a central hub, meaning that text must maintain a high degree of relevance with both formulas and images. Only when there is a significant association between text and formulas, as well as between text and images, can it be considered that there is a correlation between formulas and images. Through the implicit association of text with formulas and images, the model can indirectly infer the similarity between formulas and images as follows:

$$s(f,i) = s(f,t) \cdot s(t,i) \tag{22}$$

The product term $s(f,i)$ participates in the gradient calculation, guiding the formula and image encoders to approach the text anchor point, providing better initialization for the direct formula–image contrast in the second stage and avoiding the blindness of random initialization.

Finally, absolute similarity optimization and relative distance constraints are combined. Through the joint contrast of the three modalities, the joint distribution of text, formula, and image in the unified semantic space is explicitly optimized, thereby achieving fine-grained cross-modal alignment. Specifically, by constraining the formula and image with text first, a close association among the three is enforced, constructing a text-centered triplet. During this process, it is required that the joint similarity between the text and the positive formula/image be significantly higher than that with the negative ones, ensuring that the model can effectively distinguish between positive and negative examples. The loss function is calculated as follows:

$$\mathcal{L}_{\text{triplet}} = \max\left(0, s\left(t,f^-\right) + s\left(t,i^-\right) - \left(s\left(t,f^+\right) - s\left(t,i^+\right)\right) + \alpha\right) \tag{23}$$

$(t,f^-) + s(t,i^-)$ represents the similarity between the anchor and the negative examples, $s(t,f^+) + s(t,i^+)$ represents the similarity between the anchor and the positive examples, and α represents the minimum margin by which the total similarity between the anchor text and the positive formula/image must exceed that with the negative ones.

After the training in the second stage, the formula and image have been aligned with the text, respectively. The formula features and image features have been stably mapped into the unified semantic space. When directly calculating the contrast loss L_{f2i} between the formula and image, the noise is significantly reduced. To further balance the correlation between modalities and enhance the stability of training, it is allowed for the formula or image to be associated with the text independently to establish a connection, thereby

reducing the dependence on text and directly aligning the formula with the image. The similarity between the formula and image is calculated through text as a bridge:

$$s(f,i) = \frac{s(f,t) + s(t,i)}{2} \tag{24}$$

Since the bridging formula $s(f,i)$ naturally balances the bidirectional association, it not only achieves bidirectional alignment between the formula and the image but also avoids the computational overhead of explicitly alternating anchor points. When the formula is taken as the anchor point, the formula features are fixed, and the image features are adjusted to be as close as possible to the positive examples and as far away as possible from the negative examples. The calculation formula is:

$$\mathcal{L}_{f2i} = -\log \frac{e^{s(f,i^+)/\tau}}{e^{s(f,i^+)/\tau} + \sum_{i^-} e^{s(f,i^-)/\tau}} \tag{25}$$

Here, $s(\cdot)$ is similarity, τ is temperature coefficient, and i^+/i^- are positive/negative image samples; this formula realizes direct formula–image alignment with reduced noise.

Finally, the contrastive loss for all modality pairs is comprehensively calculated as:

$$\mathcal{L}_{joint} = \mathcal{L}_{t2f} + \mathcal{L}_{t2i} + \mathcal{L}_{f2i} \tag{26}$$

Here, $\mathcal{L}_{t2f}$, $\mathcal{L}_{t2i}$, $\mathcal{L}_{f2i}$ are text–formula, text–image and formula–image contrastive losses, respectively; this formula comprehensively optimizes multimodal joint distribution for global alignment.

Although multimodal features optimized through contrastive learning have achieved global semantic alignment, there may still be issues of imprecise matching at the local fine-grained level. To address this, a phased cross-modal fusion model is proposed to achieve hierarchical modeling of geometric relations through progressive feature interaction among text, formula, and image. The intra-modal self-attention networks enhance key features and the cross-attention mechanisms are used to fuse text and formula features. The text and formulas usually have strong semantic associations, so fusing these two first helps the model better understand the correspondence between mathematical symbols and natural language descriptions. Then, the fused text–formula features are further fused with image features to obtain the multimodal fusion features.

The detailed process is as follows: First, a self-attention network with six stacked layers is constructed [44], and TFI features are input, respectively, with residual connections and layer normalization operations introduced to ensure training stability. This network can effectively suppress redundant description information in the problem text, make the expression of formula features more precise, and enhance the feature expression of key geometric conditions through iterative updates of feature weights layer by layer. The self-attention calculation process can be expressed as:

$$Attention(Q,K,V) = Softmax(\frac{QK^\top}{\sqrt{d_k}}) \tag{27}$$

Among them, Q, K, and V respectively represent the query, key, and value. Redundant information is filtered through layer-by-layer weight updates, where d_k is the dimension of the key vector, and $QK^\top$ calculates the similarity between the query and the key.

In geometry problems, the mathematical symbols in the formula and the descriptions in the text are often closely related. By fusing text and formulas, the model can establish an accurate correspondence between symbols and semantics. This prefusion approach helps the model capture the precise matching between text and formulas, thereby reducing

the complexity of the image fusion stage. A bidirectional geometric constraint-based text–formula fusion module is constructed to establish an accurate correspondence between the formula and the text. The cross-attention calculations are performed on the enhanced text feature H_t and formula feature H_f:

$$CrossAttn_{t2f} = Softmax\left(\frac{(H_tW_q)\left(H_fW_k\right)^1}{\sqrt{d}}\right)H_fW_v \tag{28}$$

Among them, $W_q, W_k, W_v \in \mathbb{R}^{d\times d}$ are learnable parameters, the temperature coefficient $\sqrt{d}$ is used to stabilize the gradient and establish an explicit association chain between text description and formula symbol.

Finally, the text–formula fusion features are fused with the image features to obtain the final multimodal fusion features:

$$MultiAttn_{fin} = Softmax\left(\frac{\left(H_{tf}W_q\right)(H_iW_k)^1}{\sqrt{d}}\right)H_iW_v \tag{29}$$

Here, H_{tf} is text–formula fused feature and H_i is image feature; this formula achieves fine-grained fusion of multimodal features and corrects local semantic deviations.

After the above-mentioned attention enhancement processing, the model outputs the optimized multimodal features, which not only retain the discriminative advantages brought by contrastive learning but also correct the local semantic deviations through fine-grained alignment, ultimately forming high-quality feature representations suitable for geometric reasoning tasks and providing high quality input for the geometry solver.

3.4. Geometry Problem Solver Model

Text, formulas and images are processed by the TFI parser to extract text, formula and image features. These features are integrated into a unified feature vector through the TFI multimodal fusion model. This fused feature vector is passed as the input to the geometry problem solver for solution. Then, a GRU network is used to focus on multimodal fusion features to generate a target operation tree, which is organized in a tree structure and generated in a preorder or postorder traversal sequence, depending on the dataset. The structure of the operation tree ensures the validity of the generated sequence. Additionally, <CLS> and <END> markers are added at the beginning and end of the tree. The operation tree encodes the complete reasoning process of geometry problem solving, and it is further converted into a standardized final solution through tree traversal, serving as the solver's output.

The generation process starts with the symbol <CLS>. The multimodal fusion features are concatenated with the <CLS> symbol and input into the GRU. At each step, the GRU dynamically combines the current hidden state (which encodes historical reasoning information, such as previously generated geometric parameters, operators, and intermediate results) with the input fusion features, to model the sequential dependencies of geometric problem solving. At each prediction step $\{y_t\}$ ($1 \leq t \leq T$), the generation task is transformed into a classification problem. During the generation process, the validity of the operation tree is used to restrict the decision space. For example, if two numbers have been generated, the next step must predict an operator, otherwise, it will result in an invalid equation. This constraint effectively improves the performance of the answer generator.

The multimodal fusion features are concatenated with the hidden state of the GRU and then passed to a linear layer with a softmax activation function to predict the probability distribution of the next formula symbol. At each time step t, the true formula symbol,

y, is concatenated with the decoder's decoding information to generate the prediction for the next step. In the final step of the generation process, the last formula symbol is concatenated with the decoding information, and the network will predict the symbol <END>, marking the end of the formula generation process. During the experiment, the model quickly experienced overfitting, and its performance did not improve significantly in the early stage. Specifically, overfitting makes the model tend to memorize the symbol sequences, operation tree structures and reasoning paths of training samples rather than learning general and transferable geometric reasoning rules. This further leads to a series of problems that degrade the solving performance: first, it causes inaccurate prediction of key formula symbols, such as wrong operators or geometric parameters, which directly affects the correctness of the subsequent operation tree; second, it results in invalid operation tree generation, for example, unreasonable operator selection or contradictory intermediate calculation results that violate geometric logic, making the tree unable to be converted into a valid solution; third, it reduces the model's ability to adapt to unseen geometric problems, leading to consistent prediction errors for new problem types and further reducing the accuracy and reliability of the final solving results. To alleviate these negative impacts caused by overfitting, we adopted the flooding strategy, which allows the model to perform a random walk at the training loss level corresponding to the empirically best performance. Therefore, the loss function was changed to $\mathcal{L}'_a(\theta)$, as shown in Formula (30):

$$\mathcal{L}'_a(\theta) = |\mathcal{L}_a(\theta) - b| + b \tag{30}$$

This formula defines the flooding loss function, which alleviates overfitting by maintaining the training loss near the optimal value, ensuring the GRU-based solver learns general geometric reasoning rules rather than memorizing training samples. Where b is the training loss when the model performs best: when the training loss is greater than b, the model continues to perform gradient descent. While when the training loss is less than b, the model performs gradient ascent, thereby maintaining the model loss at b.

4. Experiments and Analysis

4.1. Datasets

The PGDP5K dataset contains 5000 image samples, among which geometric instances are classified into three categories: points, lines, and circles, Non-geometric instances include 22 symbol types and 6 text types, and the text is regarded as a special record in the symbols. From this dataset, we selected some representative non-geometric instances as the dataset for geometric image processing. The number of symbols in each category is shown in Table 1.

Table 1. Number of symbols of each type.

Name	Angle	Short Line	Parallel	Vertical	Text
Number	1952	2301	799	2467	26,158

To verify the effectiveness of the geometric image processing method, the GeoEval benchmark was selected for validation in this experiment [45]. This benchmark consists of diverse geometry problems sourced from seven state-of-the-art datasets, covering a wide range of geometric shapes and thus providing a comprehensive overview of various geometric mathematical challenges. Its core subsets include GeoEval-2000, GeoEval-backward, GeoEval-aug, and GeoEval-hard. Among them, GeoEval-hard was excluded from the experiment because it contains a large number of solid geometry problems, which are inconsistent with the research focus on plane geometry in this study. Therefore, three

subsets, GeoEval-2000, GeoEval-backward, and GeoEval-aug, dominated by plane geometry problems, were selected as the experimental data sources. Specifically, experimental data were extracted from the three selected subsets. Notably, all selected data perfectly matches the input requirements of our proposed geometry problem-solving method. To ensure reliable evaluation results and rational data utilization, this study adopts a random splitting strategy to divide the data, with 3500 samples for the training set and 500 samples for the testing set. The determination of this splitting ratio and scale is based on the following considerations: First, it adapts to the original scales of the three subsets and balances the sample contribution of each subset. Second, some samples with poor adaptability are screened out during the splitting process to ensure data quality. Third, 3500 training samples are sufficient to support the model in learning key geometric features and reasoning logics, avoiding underfitting caused by insufficient data volume. Fourth, 500 testing samples meet the requirements of statistical significance, which can reliably verify the generalization ability of the model without generating excessive computational overhead. The random splitting strategy can also ensure unbiased data distribution, effectively avoiding evaluation deviations caused by fixed partitioning. The detailed statistical results of data partitioning are shown in Table 2.

Table 2. GeoEval Statistics.

Category	GeoEval-2000	GeoEval-Backward	GeoEval-Aug
Training Set	1500	500	1500
Testing Set	200	100	200

4.2. Experimental Details

In this work, Python is chosen as the programming language and Pytorch framework is used to implement the model. The CUDA 11.1 version environment is installed and the A6000 GPU is used for accelerating computation. To optimize the training process of the model, the learning rate is set to 1×10^{-3}, the batch size is set to 32, Adam optimizer is used, and the epochs are set to 100. The backbone adopts MobileNetV2. The non-geometric detection module and the geometric segmentation module both use three groups of 128-channel convolution layers and corresponding BatchNorm layers.

4.3. Experimental Results

To verify the effectiveness of our method in detecting geometric instances and non-geometric instances, we compare our method with SSD [46], RetinaNet [8] and Mask R-CNN [19] on the same dataset for non-geometric instance detection. The results are shown in Table 3.

Table 3. Comparison of detection accuracy for non-geometric instances.

Method	Angle (%)	Short Line (%)	Parallel (%)	Vertical (%)	Text (%)
SSD	46.99	79.28	84.83	95.99	98.01
RetinaNet	47.56	77.49	86.35	93.85	98.52
Mask R-CNN	53.26	83.59	90.34	97.25	99.03
Ours	86.27	96.71	95.03	99.79	99.37

As shown in Table 3, our method has significant improvements in many aspects compared with the SSD, RetinaNet and Mask R-CNN, which indicates that our method shows certain superiority in the detection of non-geometric instances.

For geometric instance detection, we test the performance of geometric instance detection on the PGDP5K dataset. Table 4 describes the results of geometric instance detection, where line instances cover all collinear segments. We can see that compared with the traditional method GEOS, our method has achieved significant improvements, especially in terms of points and lines.

Table 4. Comparison of detection accuracy of geometric examples.

Method	Point (%)	Line (%)	Circle (%)
GEOS	76.51	66.99	95.25
Ours	99.25	99.35	99.86

In order to evaluate the effectiveness of the proposed geometry problem solver, experiments were conducted on the GeoEval-2000 dataset, and the performance results of solving geometry problems were obtained. The accuracy was used as the evaluation index to measure the proportion of the number of correctly predicted questions to the total number of questions by the model. It reflects the accuracy of the model's prediction. Figure 5 includes the solving results of WizardMath-70B, WizardMath-7B-V1.1 [47], GPT-4V [48] and our method.

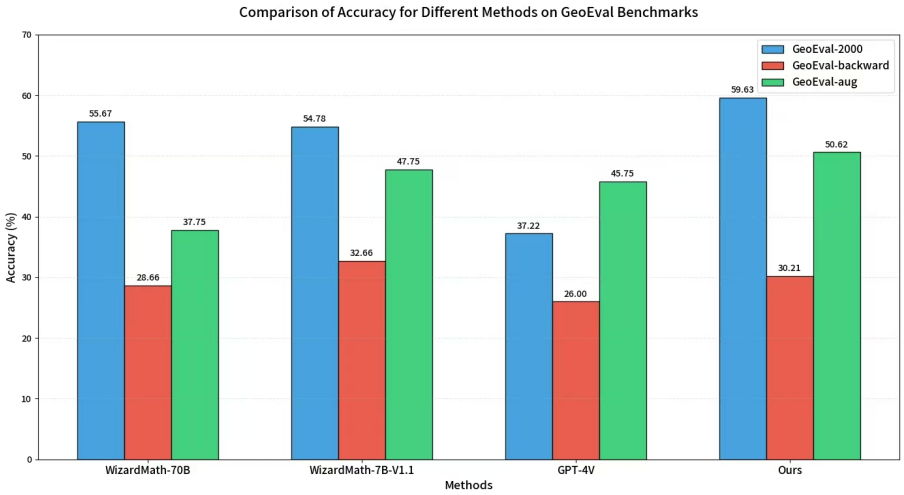


Figure 5. Comparison of accuracy for different methods on GeoEval.

As shown in Figure 5, the proposed method was compared with three baseline models on three GeoEval subsets. The results confirm the method's effectiveness in geometry problem solving, with detailed analysis as follows:

(1) In the standard subset GeoEval-2000, our method achieves 59.63% accuracy, outperforming WizardMath-70B, WizardMath-70B-V1.1, and GPT-4V by 3.96%, 4.85%, and 22.41%, respectively. This indicates its strong basic geometry solving capability. In the data-augmented subset GeoEval-aug, our method maintains leading performance, outperforming WizardMath-70B-V1.1 by 2.87%. This validates its excellent generalization.

(2) In the backward reasoning subset GeoEval-backward, our method achieves 30.21% accuracy, outperforming WizardMath-70B and GPT-4V, though slightly lower than WizardMath-70B-V1.1. Backward reasoning is inherently challenging, leading to overall low accuracy across models. Our competitive results demonstrate that TFI fusion features effectively support reverse logical reasoning.

In summary, our method outperforms baselines in standard tasks, demonstrates excellent generalization in augmented scenarios, and achieves competitive results in backward

reasoning. These results initially validate the effectiveness of the proposed TFI multimodal parsing and fusion model, and the specific correlation between the experimental data and the three core contributions of the paper will be discussed in detail in the following section. This fully validates the effectiveness of the proposed geometry problem solver. Our model slightly underperforms WizardMath-7B-V1.1 on GeoEval-backward. Key reasons are that backward reasoning requires reverse logic which WizardMath-7B-V1.1 is specifically optimized for via mathematical pre-training our model prioritizes multimodal fusion for geometric perception over pure backward deduction and WizardMath-7B-V1.1 allocates more parameters to reasoning with no multimodal overhead. Despite this our model remains competitive on this subset. We will optimize backward reasoning in future work while retaining multimodal advantages.

4.4. Ablation Experiment

4.4.1. Ablation Experiments on Multimodal Fusion and Formula Parser Contribution

In this section, the effects of MFB model and fine-grained multimodal fusion model on the solution accuracy will be evaluated on the GeoEval-2000 dataset through multimodal fusion ablation experiments.

The training effect is shown in Figure 6. We take the method without multimodal fusion as the baseline and compared it with the MFB fusion and the fine-grained fusion methods. The experimental results show that in the initial stage, the model's Loss value decreased rapidly, and then the rate of decrease tended to be stable, and it stabilized after 80 training rounds. Specifically, the loss of the MFB fusion method decreases significantly faster than that of the baseline method, while the fine-grained fusion method has the fastest decrease rate, indicating that it is the most efficient in feature learning. This further validates the superiority of the MFB fusion and fine-grained fusion methods, among which the fine-grained fusion method had the most significant effect.

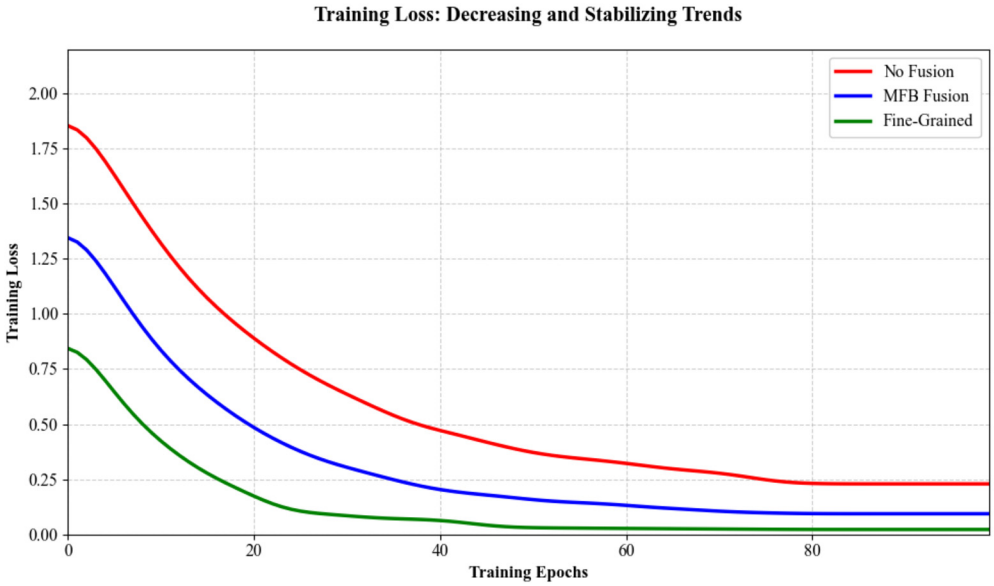


Figure 6. Curve of loss value variation.

As shown in Table 5, experimental results show that in the absence of multimodal fusion, although the TFI features are used in the model at the same time, the solution accuracy is only 50.26% due to the failure to integrate these three features effectively. This shows that the TFI feature processing alone cannot fully utilize the multidimensional information of the problem, limiting the accuracy of the solution. The MFB model is used to fuse the TFI features, and the solution accuracy is improved to 54.62%, demonstrating

the importance of multimodal fusion in geometry problem solving. After using the fine-grained multimodal fusion model, the TFI features are refined through the multimodal contrast model, which makes the distance of relevant samples in the shared space closer, and the solution accuracy is further improved to 59.63%. To further verify the contribution of the formula parser and the formula modality, we additionally design a group of contrast experiments using only text and image bimodal features without formula information. We set up three parallel configurations: text and image without fusion, text and image with MFB fusion, and text and image with our fine-grained fusion. Experimental results illustrate that all bimodal schemes achieve significantly lower accuracy than the corresponding TFI trimodal schemes. This gap clearly proves that the formula modality processed by the formula parser provides unique mathematical logic and geometric constraints, which cannot be replaced by text and image modalities. Taken together, the model significantly improves the overall problem-solving performance by effectively integrating TFI features, and makes progress in several key areas, emphasizing the importance of multimodal information in complex problem solving.

Table 5. Comparison of accuracy of multimodal fusion methods.

Method	TFI	Text + Image
No fusion	50.26	45.18
MFB	54.62	49.56
Ours	59.63	52.36

4.4.2. Ablation Experiment on Core Modules of Image Parser

To further verify the necessity of each core module in the geometric image parser and explore the feasibility of model simplification, we conducted ablation experiments on the PGDP5K dataset and the non-geometric instance dataset. The experiments focus on three key modules of the image parser: FPN for multiscale feature extraction, FCOS for anchor-free non-geometric detection, and the dual-branch geometric segmentation module that integrates semantic segmentation and embedding segmentation.

We took the complete geometric image parser as the benchmark model and compared it with models under different module configurations. The main evaluation indicator includes the detection accuracy of each category.

First, we tested the model with FPN removed. After removing FPN, this model adopts a single-scale feature extraction method while retaining other core modules. The experimental results show that the detection accuracy of this model is significantly lower than that of the benchmark model: for non-geometric instances, the angle detection accuracy drops from 86.27% to 62.35%, and the short line detection accuracy drops from 96.71% to 85.12%; for geometric instances, the point detection accuracy decreases from 99.25% to 82.67%, and the line detection accuracy decreases from 99.35% to 75.32%. This indicates that FPN plays a key role in extracting features of multiscale instances such as small symbols and slender lines in geometric images; removing FPN will sacrifice detection accuracy.

Second, we replaced the anchor-free FCOS non-geometric detection module with SSD. The experimental results show that the non-geometric instance detection accuracy of the modified model is significantly lower than that of the benchmark model. Among them, the angle detection accuracy is only 51.42%, which is close to 46.99% of SSD in Table 3 and much lower than 86.27% of the benchmark model. This fully proves that compared with traditional methods such as SSD, FCOS is more suitable for non-geometric instance detection in geometry problem scenarios. The detailed results of the above ablation experiments are shown in Table 6.

Table 6. Ablation experiment results of image parser core modules.

Model Configuration	Non-Geometric Detection Accuracy (%)					Geometric Detection Accuracy (%)		
	Angle	Short Line	Parallel	Vertical	Text	Point	Line	Circle
Without FPN	62.35	85.12	88.47	97.21	98.5	82.67	75.32	95.01
FCOS replaced with SSD	51.42	80.36	86.19	96.58	98.21	99.25	99.35	99.86
Ours	86.27	96.71	95.03	99.79	99.37	99.25	99.35	99.86

In summary, the ablation experiments show that FPN, FCOS are all core components that ensure the high detection accuracy of the geometric image parser. The complex configuration of these modules is not redundant but tailored for the characteristics of geometry problem images such as multiscale instances, slender and overlapping geometric primitives.

4.5. Result Discussion and Contribution Validation

The experimental results above correlate closely with the three core contributions of this paper. Specific quantitative data fully verify the effectiveness of each designed module and method, further confirming the rationality and advancement of the proposed TFI multimodal parsing and fusion model for geometry problem solving.

For the dedicated formula parser optimized for mathematical formula structures, the ablation results in Table 5 demonstrate that the TFI trimodal fusion with formula features processed by our parser outperforms the text–image bimodal fusion under all settings. The accuracy of our fine-grained fusion scheme reaches 59.63%, which is obviously higher than 52.36% of the bimodal counterpart without formula information. Such a promotion fully proves that the formula parser effectively enhances formula feature extraction and structural representation, providing unique mathematical and logical constraints that cannot be supplied by text and image modalities alone. It thus lays a solid foundation for subsequent cross-modal fusion and steadily boosts the performance of the entire geometry problem-solving model.

For the geometric image parser framework for geometric image instance detection and segmentation, Tables 3 and 4 and the ablation experiment results in Table 6 provide comprehensive quantitative verification. Compared with traditional SSD, RetinaNet, Mask R-CNN and GEOS, our method greatly improves the detection accuracy of non-geometric and geometric instances as shown in Tables 3 and 4. The ablation experiments in Table 6 further confirm that FPN and FCOS are core components for ensuring high detection accuracy—removing or simplifying any of these modules will lead to significant accuracy degradation, even though computational cost can be reduced. This demonstrates that the framework's complex configuration is tailored for the characteristics of geometry problem images such as multiscale instances and slender/overlapping primitives, effectively overcoming the recognition bottlenecks of traditional methods. Meanwhile, the simplified models in the ablation experiment also prove the feasibility of lightweight alternatives, which can be applied according to actual accuracy and computational cost requirements.

For the TFI multimodal fusion model based on geometric semantic constraints, Figure 5 and Table 5 form a joint verification system. The ablation experiment in Table 5 shows the fine-grained fusion model with text rewriting and cross-attention mechanisms outperforms the non-fusion baseline and MFB fusion method, verifying that the fusion strategy optimizes text–formula–image feature alignment and enhances semantic consistency and feature expressiveness. The comparison results in Figure 5 show our model reaches 59.63% accuracy in GeoEval-2000, outperforming WizardMath-70B by 3.96 percentage

points. This proves hierarchical attention-based precise TFI fusion reduces redundant information, highlights key geometric features, and the generated high-quality fusion vectors significantly boost the model's overall geometry problem-solving ability. But it is worth noting that the proposed method also has potential limitations in backward reasoning tasks. As reflected in the GeoEval-backward subset experiment, the model's accuracy is slightly lower than that of WizardMath-7B-V1.1, mainly due to the fact that the model's core optimization focuses on multimodal feature fusion for forward reasoning, while lacking targeted optimization for reverse logical deduction that relies more on mathematical axiom accumulation. This limitation indicates that there is room for improvement in the model's ability to handle complex backward reasoning scenarios.

In summary, all experimental results align with the paper's initial contribution assumptions. The formula parser, image parser and TFI fusion model all show significant performance improvement effects, fully demonstrating the effectiveness and innovation of the proposed method.

5. Conclusions

This study sought to find a solution to the problem in which traditional text feature extraction methods cannot effectively extract formula information. The aim was to resolve the deficiency of traditional image feature extraction methods in terms of accuracy, as well as the insufficient fusion of the geometric TFI features. This paper proposes a method of solving geometry problems based on a TFI multimodal parsing and fusion model. In addition to a dedicated text parser, a formula feature extractor has also been specially designed to extract features from the formula parts in the questions, improving the overall accuracy. In the image parser, by differentiating the processing of geometric instances and non-geometric instances, the problems of blurred segmentation of slender geometric bodies and inaccurate positioning of small-sized symbols are solved, thereby improving the extraction effect of image features. Then, this paper introduces the multimodal fusion model and combines the contrastive learning mechanism. By stacking self-attention and cross-attention to construct deep feature expressions, it fully integrates TFI features, solving the problem that TFI features cannot be effectively fused. Finally, the GRU network utilizes the attention mechanism to process multimodal fusion features, generates the target tree, and generates the corresponding geometry solutions. The proposed method performs excellently in standard geometry problem-solving tasks and exhibits strong generalization ability in data-augmented scenarios, outperforming all baseline models in these two subsets. In backward reasoning tasks, the method also achieves competitive performance by outperforming WizardMath-70B and GPT-4V, though its accuracy is slightly lower than WizardMath-70B-V1.1. In addition, multimodal fusion ablation experiments are also carried out. The ablation experiment verifies the effective fusion of TFI features, significantly improving the accuracy of solving geometry problems. These results indicate that making full use of multimodal information is of great significance for improving the accuracy of solving geometry problems. This method has obvious advantages and provides valuable references for future research.

Author Contributions: Conceptualization, P.J. and T.S.; methodology, P.J., Z.S., T.S. and Y.W.; software, Z.S.; validation, P.J., Z.S., T.S. and Y.W.; formal analysis, Z.S.; investigation, Z.S.; resources, P.J.; data curation, Z.S.; writing—original draft preparation, Z.S.; writing—review and editing, P.J.; visualization, Z.S.; supervision, P.J.; project administration, P.J. and Y.W.; funding acquisition, P.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Natural Science Foundation of China (No. 62207010, 62107014), the General Project of Natural Science Foundation of Henan Province (262300421801), the Soft Science Project of Henan Province (No. 262400410529), and the Science and Technology Innovation Teams in Universities of Henan Province (No. 23A520002).

Data Availability Statement: The data presented in this study are available within the article. The datasets analyzed during the current study (PGDP5K and GeoEval) are publicly available resources in PGDP5K-And-GeoEval at [https://github.com/hhhh0805/PGDP5K-And-GeoEval], accessed on 3 January 2026, and are cited accordingly in the manuscript.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Zhu, N.; Zhang, X.K.; Huang, Q.K.; Zhu, F.Z.; Zeng, Z.B.; Leng, T. FGeo-Parser: Autoformalization and Solution of Plane Geometric Problems. *Symmetry* **2024**, *17*, 8. [CrossRef]
2. Sachan, M.; Dubey, A.; Hovy, E.H.; Mitchell, T.M.; Roth, D.; Xing, E.P. Discourse in multimedia: A Case Study in Extracting Geometry Knowledge from Text-books. *Comput. Linguist.* **2019**, *45*, 627–665. [CrossRef]
3. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*; Volume 1 (Long and Short Papers); Association for Computational Linguistics (ACL): Stroudsburg, PA, USA, 2019; pp. 4171–4186.
4. Zhang, X.; Cao, J.T.; Wei, J.Q.; Xu, Y.W.; You, C.Y. Tokenization Constraints in LLMs: A Study of Symbolic and Arithmetic Reasoning Limits. *arXiv* **2025**, arXiv:2505.14178. [CrossRef]
5. Liu, H.; Cui, C.; Du, Y.N.; Liu, Y.; Pan, G. PP-FormulaNet: Bridging Accuracy and Efficiency in Advanced Formula Recognition. *arXiv* **2025**, arXiv:2503.18382.
6. Li, Z.; Wang, X.Y.; Liu, Y.L.; Jin, L.W.; Huang, Y.C.; Ding, K. Improving Handwritten Mathematical Expression Recognition via Similar Symbol Distinguishing. *IEEE Trans. Multimed.* **2024**, *26*, 90–102. [CrossRef]
7. He, K.M.; Zhang, X.Y.; Ren, S.Q.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
8. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.M.; Dollár, P. Focal Loss for Dense Object Detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988.
9. Jia, C.; Yang, Y.F.; Xia, Y.; Chen, Y.T.; Parekh, Z.; Pham, H.; Le, Q.V.; Sung, Y.; Li, Z.; Duerig, T. Scaling Up Visual and Vision-Language Representation Learning with Noisy Text Supervision. In Proceedings of the 38th International Conference on Machine Learning, Virtual Event, 18–24 July 2021; pp. 4904–4916.
10. He, K.M.; Fan, H.Q.; Wu, Y.X.; Xie, S.N.; Girshick, R. Momentum Contrast for Unsupervised Visual Representation Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 9729–9738.
11. Zhang, M.L.; Yin, F.; Liu, C.L. A Multi-Modal Neural Geometric Solver with Textual Clauses Parsed from Diagram. *arXiv* **2023**, arXiv:2302.11097. [CrossRef]
12. Lu, P.; Gong, R.; Jiang, S.; Qiu, L.; Huang, S.Y.; Liang, X.D.; Zhu, S.C. Inter-gps: Interpretable Geometry Problem Solving with Formal Language and Symbolic Reasoning. In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), Online, 1–6 August 2021.
13. Zhang, M.L.; Yin, F.; Hao, Y.H.; Liu, C.L. Plane Geometry Image Parsing. In Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, Vienna, Austria, 23–29 July 2022; pp. 1636–1643.
14. Ping, B.W.; Luo, M.N.; Dang, Z.H.; Wang, C.X.; Jia, C.Y. AutoGPS: Automated Geometry Problem Solving via Multimodal Formalization and Deductive Reasoning. *arXiv* **2025**, arXiv:2505.23381. [CrossRef]
15. Chen, G.Z.; Xu, W.W.; Zhang, H.; Chan, H.P.; Zhao, D.; Luu, A.T.; Rong, Y. GeoPQA: Bridging the Visual Perception Gap in MLLMs for Geometric Reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2025*; Association for Computational Linguistics: Suzhou, China, 2025; pp. 25680–25688.
16. Li, Z.L.; Zhang, W.X.; Yan, C.; Zhou, Q.Y.; Li, C.; Liu, H.Z.; Cao, Y.B. Seeking Patterns, Not just Memorizing Procedures: Contrastive Learning for Solving Math Word Problems. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, 22–27 May 2022; pp. 2486–2496.

17. Liang, Z.W.; Zhang, J.P.; Wang, L.; Qin, W.; Lan, Y.S.; Shao, J.; Zhang, X.L. MWP-BERT: Numeracy-Augmented Pre-training for Math Word Problem Solving. In Proceedings of the Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, USA, 10–15 July 2022; pp. 997–1009.
18. Jian, P.P.; Sun, T.H.; He, B. A Generation Model of Math Word Problem Solution via Kernel Mean Dynamic Aggregation and Dual-Decoder Synergy Fusion. *Inf. Fusion* **2026**, *127*, 103826. [CrossRef]
19. He, K.M.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2961–2969.
20. Ying, H.; Huang, Z.J.; Liu, S.; Shao, T.J.; Zhou, K. Embedmask: Embedding coupling for one-stage instance segmentation. In Proceedings of the IJCAI, Montreal, QC, Canada, 19–26 August 2021.
21. Yuan, Y.Q.; Cheng, Y.; Pan, B.; Jin, G.; Yu, D.; Ye, M.J.; Zhang, Q. A Multi-Modal Attention Fusion Framework for Road Connectivity Enhancement in Remote Sensing Imagery. *Mathematics* **2025**, *13*, 3266. [CrossRef]
22. Jian, P.P.; Guo, F.C.; Wang, Y.L.; Li, Y. Solving Geometry Problems via Feature Learning and Contrastive Learning of Multi-Modal Data. *Comput. Model. Eng. Sci.* **2023**, *136*, 1707–1728.
23. Li, L.H.; Yatskar, M.; Yin, D.; Hsieh, C.J.; Chang, K.W. VisualBERT: A Simple and Performant Baseline for Vision and Language. *arXiv* **2019**, arXiv:1908.03557. [CrossRef]
24. Jian, P.P.; Guo, F.C.; Pan, C.; Wang, Y.L.; Yang, Y.R.; Li, Y. Interpretable Geometry Problem Solving Using Improved RetinaNet and Graph Convolutional Network. *Electronics* **2023**, *12*, 4578. [CrossRef]
25. Jian, P.P.; Zhang, X.H.; Wu, L.; Ma, B.; Hong, W.Y. Geometry Diagram Parsing and Reasoning Based on Deep Semantic Fusion. *Symmetry* **2026**, *18*, 92. [CrossRef]
26. Peng, S.; Fu, D.; Liang, Y.J.; Gao, L.C.; Tang, Z. GeoDRL: A Self-Learning Framework for Geometry Problem Solving using Reinforcement Learning in Deductive Reasoning. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2023, Toronto, ON, Canada, 9–14 July 2023; pp. 13468–13480.
27. Lin, T.Y.; Dollar, P.; Girshick, R.; He, K.M.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
28. Tian, Z.; Shen, C.H.; Chen, H.; He, T. FCOS: A Simple and Strong Anchor-Free Object Detector. In Proceedings of the IEEE Transactions on Pattern Analysis and Machine Intelligence 2022, Piscataway, NJ, USA, 1 April 2022; Volume 44, pp. 1922–1933.
29. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
30. Zhang, H.; Koh, J.Y.; Baldridge, J.; Lee, H.; Yang, Y.F. Cross-Modal Contrastive Learning for Text-to-Image Generation. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 833–842.
31. Chen, J.; Jing, H.; Chang, Y.; Liu, Q. Gated Recurrent Unit Based Recurrent Neural Network for Remaining Useful Life Prediction of Nonlinear Deterioration Process. *Reliab. Eng. Syst. Saf.* **2019**, *185*, 372–382. [CrossRef]
32. Joshi, M.; Chen, D.Q.; Liu, Y.H.; Weld, D.S.; Zettlemoyer, L.; Levy, O. Spanbert: Improving Pretraining by Representing and Predicting Spans. *Trans. Assoc. Comput. Linguist.* **2020**, *8*, 64–77.
33. Li, W.; Gao, C.; Niu, G.C.; Xiao, X.Y.; Liu, H.; Liu, J.C.; Wu, H.; Wang, H.F. UNIMO: Towards Unified-Modal Understanding and Generation via Cross-Modal Contrastive Learning. In Proceedings of the ACL-IJCNLP 2021—59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Online, 1–6 August 2021; pp. 2592–2607.
34. Yu, X.G.; Wang, M.S.; Gan, W.B.; He, B.; Ye, N. A Framework for Solving Explicit Arithmetic Word Problems and Proving Plane Geometry Theorems. *Int. J. Pattern Recognit. Artif. Intell.* **2019**, *33*, 1950016. [CrossRef]
35. Jian, P.P.; Sun, C.; Yu, X.G.; He, B.; Xia, M. An End-to-End Algorithm for Solving Circuit Problems. *Int. J. Pattern Recognit. Artif. Intell.* **2019**, *33*, 1940004. [CrossRef]
36. Scharpf, P.; Schubotz, M.; Cohl, H.S.; Breitinger, C.; Gipp, B. Discovery and Recognition of Formula Concepts Using Machine Learning. *Scientometrics* **2023**, *128*, 4971–5025. [CrossRef]
37. Chen, Y.; Du, J.; Xue, M.B.; Zhang, J.S. An Encoder-Decoder Based Generation Model for Online Handwritten Mathematical Expressions. *J. Image Graph.* **2023**, *28*, 2356–2369.
38. Zhang, W.Q.; Shen, Y.L.; Nong, Q.P.; Tan, Z.Q.; Ma, Y.N.; Lu, W.M. An Expression Tree Decoding Strategy for Mathematical Equation Generation. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP), Singapore, 6–10 December 2023; pp. 439–456.
39. Xia, R.Q.; Li, M.S.; Ye, H.C.; Wu, W.J.; Zhou, H.B.; Yuan, J.K.; Peng, T.S.; Cai, X.Y.; Yan, X.C.; Wang, B.; et al. GeoX: Geometric Problem Solving Through Unified Formalized Vision-Language Pre-training. *arXiv* **2025**, arXiv:2412.11863.
40. Gao, J.H.; Pi, R.J.; Zhang, J.P.; Ye, J.C.; Zhong, W.J.; Wang, Y.F.; Hong, L.Q.; Han, J.H.; Xu, H.; Li, Z.G.; et al. G-LLaVA: Solving Geometric Problem with Multi-Modal Large Language Model. *arXiv* **2025**, arXiv:2312.11370.

41. Cangea, C.; Veličković, P.; Jovanović, N.; Kipf, T.; Liò, P. Towards Sparse Hierarchical Graph Classifiers. *arXiv* **2018**, arXiv:1811.01287. [CrossRef]
42. De Brabandere, B.; Neven, D.; Van Gool, L. Semantic Instance Segmentation with A Discriminative Loss Function. *arXiv* **2017**, arXiv:1708.02551. [CrossRef]
43. Neven, D.; De Brabandere, B.; Georgoulis, S.; Proesmans, M.; Gool, L.V. Towards End-to-End Lane Detection: An Instance Segmentation Approach. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Suzhou, China, 26–30 June 2018; pp. 286–291.
44. Yu, Z.; Yu, J.; Cui, Y.H.; Tao, D.C.; Tian, Q. Deep Modular Co-attention Networks for Visual Question Answering. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 6281–6290.
45. Zhang, J.X.; Li, Z.Z.; Zhang, M.L.; Yin, F.; Liu, C.L.; Moshfeghi, Y. GeoEval: Benchmark for Evaluating LLMs and Multi-Modal Models on Geometry Problem-Solving. In Proceedings of the Findings of the Association for Computational Linguistics (ACL 2024), Bangkok, Thailand, 11–16 August 2024; pp. 1–14.
46. You, S.; Bi, Q.; Ji, Y.; Liu, S.; Feng, Y.; Wu, F. Traffic Sign Detection Method Based on Improved SSD. *Information* **2020**, *11*, 475. [CrossRef]
47. Yu, L.H.; Jiang, W.S.; Shi, H.; Yu, J.C.; Liu, Z.Y.; Zhang, Y.; Kwok, J.T.; Li, Z.G.; Weller, A.; Liu, W.Y. Metamath: Bootstrap Your Own Mathematical Questions for Large Language Models. *arXiv* **2023**, arXiv:2309.12284. [CrossRef]
48. OpenAI GPT-4 Technical Report. *arXiv* **2023**, arXiv:2303.08774.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.