Image Over Text: Transforming Formula Recognition Evaluation with Character Detection Matching

Bin Wang[1*] Fan Wu[1*] Linke Ouyang[1*] Zhuangcheng Gu[1]
Rui Zhang[1] Renqiu Xia[1,2] Botian Shi[1] Bo Zhang[1] Conghui He[1†]

[1]Shanghai Artificial Intelligence Laboratory [2]Shanghai Jiao Tong University

Abstract

Formula recognition presents significant challenges due to the complicated structure and varied notation of mathematical expressions. Despite continuous advancements in formula recognition models, the evaluation metrics employed by these models, such as BLEU and Edit Distance, still exhibit notable limitations. They overlook the fact that the same formula has diverse representations and is highly sensitive to the distribution of training data, thereby causing unfairness in formula recognition evaluation. To this end, we propose a Character Detection Matching (CDM) metric, ensuring the evaluation objectivity by designing an image-level rather than a LaTeX-level metric score. Specifically, CDM renders both the model-predicted LaTeX and the ground-truth LaTeX formulas into image-formatted formulas, then employs visual feature extraction and localization techniques for precise character-level matching, incorporating spatial position information. Such a spatially-aware and character-matching method offers a more accurate and equitable evaluation compared with previous BLEU and Edit Distance metrics that rely solely on text-based character matching. Experimentally, we evaluated various formula recognition models using CDM, BLEU, and ExpRate metrics. Their results demonstrate that the CDM aligns more closely with human evaluation standards and provides a fairer comparison across different models by eliminating discrepancies caused by diverse formula representations. Code is available at https://github.com/opendatalab/UniMERNet/tree/main/cdm

1. Introduction

Mathematical formula recognition is crucial in document analysis as it directly impacts the scientific rigor and accuracy of the document content [7, 19, 31, 37]. Unlike standard Optical Character Recognition (OCR) technique, formula recognition presents unique challenges. Formulas often encompass multi-level symbols, subscripts, fractions, and other complicated structures, requiring models to comprehend spatial and structural relationships rather than just linear, sequential text [14, 19]. Besides, formulas exhibit representational diversity, meaning that the same formula can be expressed in multiple valid ways.

In recent years, significant advancements in formula recognition [4, 8, 20, 28, 39, 41, 42] have been primarily driven by the continuous development of artificial intelligence technique [5, 35, 40]. Besides, commercial formula recognition software like Mathpix[1] and the recently proposed UniMERNet [30] model have achieved impressive results in diverse real-world settings. Despite these advancements, the existing evaluation metrics [17, 27] for formula recognition still face some challenges. Commonly-used metrics such as BLEU [27] and Edit Distance [17] primarily rely on text-based character matching, which introduces several limitations as follows:

(1) Low Metric Reliability. BLEU and Edit Distance are reliable for evaluating the quality of text-level similarity. However, the diversity in formula representations makes these text-level evaluation metrics inadequate for precisely reflecting formula recognition quality. For example, as shown in Figure 1 (Case 1), a model's prediction might render an image identical to the ground truth formula. However, due to the variations in formula expression styles, the evaluation results obtained using the ExpRate [8], BLEU, and Edit Distance may be somewhat misleading.

(2) Unfair Model Comparison. Current metrics may be susceptible to discrepancies between the distributions of training and testing data. As illustrated in Figure 1 (Case 1 and Case 2), a model may produce a correct prediction but score poorly due to representational differences from the ground truth, while an incorrect prediction might score higher if its representation aligns more closely with the test

*Equal contribution. †Corresponding author: heconghui@pjlab.org.cn

[1]https://mathpix.com/equation-to-latex

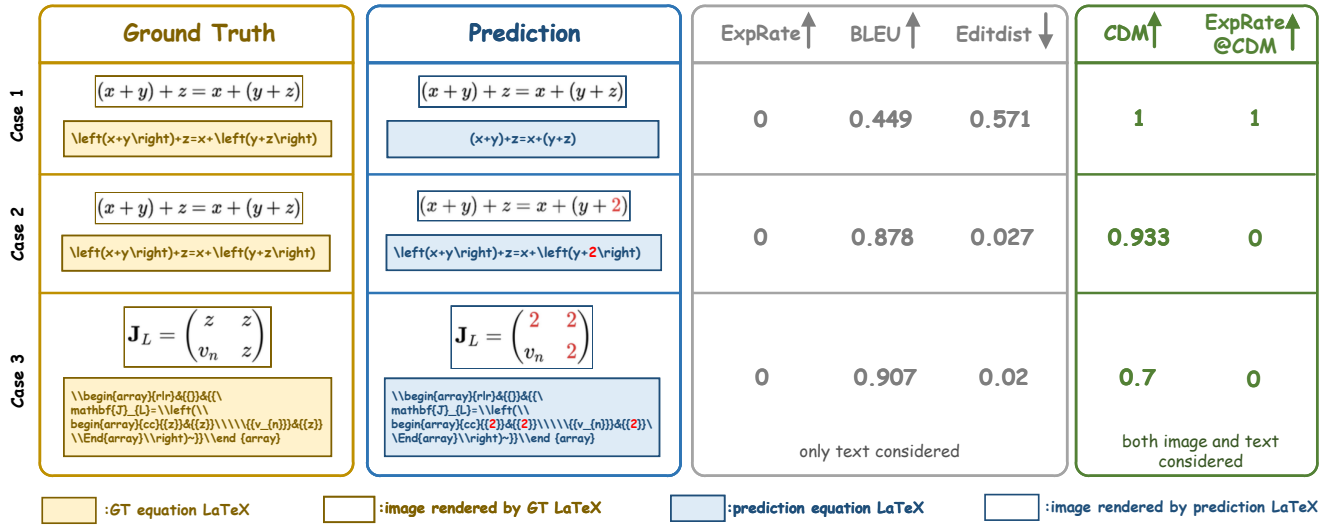


Figure 1. Illustration of the limitations of the existing metrics (ExpRate [8], BLEU [27], and Edit Distance [17]) and the advantages of the proposed CDM. **Case 1:** Due to different expression styles, the model prediction and Ground Truth (GT) may appear visually similar, but they receive low scores when evaluated using traditional metrics such as BLEU. **Case 2:** Errors in numeric predictions within formulas are essentially prediction errors. However, when evaluated using BLEU, such errors may receive higher metric scores due to the model predicting an expression style close to the GT. **Case 3:** Model predictions that are visually obviously incorrect may receive high scores using BLEU metric, which is inconsistent with human judgment standards.

data distribution.

(3) Lack of Intuitive Scoring. There can be a significant discrepancy between BLEU scores and human perception. For instance, in Figure 1 (Case 3), a model's prediction contains many errors, yet the BLEU score is as high as 0.907, which does not align with human judgment.

To address these issues, we propose a novel evaluation metric for formula recognition: Character Detection Matching (CDM). The proposed CDM regards the formula recognition evaluation as an image-based object detection task, by converting both the predicted LaTeX and the ground-truth LaTeX formulas into the image-formatted formulas and treating each character as an independent target. This approach overcomes the challenges posed by the diverse expression styles of formulas and aligns more closely with human subjective evaluation standards. CDM offers the advantages as follows: **1) Accuracy and Reliable.** By calculating metrics in the image space, CDM eliminates issues caused by different valid representations of the same formula, directly reflecting recognition accuracy and aligning more closely with human intuitive perception. **2) Fairness.** CDM removes the high dependency on consistent data distribution between training and evaluation task, allowing for a fair comparison of different models based on their true recognition capabilities. Our contributions can be summarized as follows:

- We perform a detailed analysis of the existing formula recognition evaluation metrics, highlighting the limitations of ExpRate and BLEU and their unreliability specifically for evaluating formula recognition tasks.
- We introduce a novel evaluation metric, CDM, which assesses formula recognition quality by performing visual character matching between rendered images of predicted and ground-truth formulas, providing an intuitive and fair evaluation standard.
- We validate CDM's effectiveness through extensive experiments on various mainstream models and datasets, demonstrating its superiority over traditional metrics like BLEU in assessing formula recognition performance.

2. Related Work

2.1. Formula Recognition Algorithms

Initially, researchers employ specific grammar rules to represent the spatial structure of formulas, including graph grammars [15], relational grammars [22], and probabilistic grammars [1, 2]. Besides, the CROHME competitions [16, 23–26] have promoted the development of handwritten formula recognition by incorporating deep learning algorithms. Key contributions include a neural encoder-decoder model with coarse-to-fine attention [8], a tree-structured decoder [42], and the Counting-Aware Network [18], which integrates a weakly-supervised counting module. The ABM network [3] employs mutual distillation and an Attention Aggregation Module, while a transformer-based decoder [43] simplifies model architecture. The Syntax-Aware Network (SAN) [41] models recognition as a tree traversal process, significantly improving accuracy for complex expressions. Overall, these models employ ExpRate [8] for formula recognition evaluation.

In document information extraction [18, 29, 37, 38], Donut [12] directly converts input documents into structured outputs without using traditional OCR tools. Texify [28] and UniMERNet [30] are designed using Donut [12], utilizing more diverse datasets and data augmentation operations. Nougat [5] is designed to convert PDF documents from screenshot to Markdown format, making the document content (*e.g.* table and formula) easier to edit. These methods use BLEU [27] and Edit Distance [17] metrics for formula recognition evaluation.

2.2. Formula Recognition Evaluation Metrics

BLEU is initially proposed for machine translation tasks, matching standard and machine-translated texts using N-grams (sequences of N words) between the generated and the reference texts. It applies a brevity penalty factor to produce the final BLEU score [27]:

$$BLEU = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right), \tag{1}$$

where BP is the brevity penalty factor, and p_n is the N-gram match result, with n ranging from 1 to 4.

Edit Distance is also commonly-used metric to assess the similarity between the generated and the reference texts. It measures the number of insertions, deletions, or substitutions needed to transform one text into another, with a smaller Edit Distance indicating higher similarity [17].

ExpRate refers to the proportion of samples where the texts are exactly matched out of the total number of samples. Compared to BLEU and Edit Distance, ExpRate is coarser and more stringent in evaluation [18].

The above three metrics can effectively evaluate the textual differences between ground truth and reference, making them suitable for tasks requiring strict matches. BLEU and Edit Distance, in particular, provide a finer evaluation of text recognition capabilities compared to ExpRate, making them widely used in extensive text recognition tasks such as document recognition [5, 11]. These metrics are also applied to formula recognition, with most open-source models, such as Pix2Tex [4] and Texify, adopting them for evaluation and comparison.

In addition to text-based metrics, image edit distance has been explored to measure the accuracy of predicted formulas [32]. Image processing metrics like MSE (Mean Squared Error) and SSIM [33] have also been considered. Structuring Chart-oriented Representation Metric (SCRM) [36] is designed to comprehensively evaluate the information represented by structured triplet representations. However, these metrics are better suited for natural images. For document images such as formula images, even slight character misalignments can result in significant penalties, making these metrics less suitable for formula recognition.

3. Limitations of Current Metrics

Although ExpRate, BLEU, and Edit Distance are widely used in formula evaluation tasks, they exhibit significant limitations in accurately reflecting formula recognition performance, particularly in scenarios where there are domain gaps between training and testing data distributions. The main reason is that a single formula can have multiple valid LaTeX representations, making the Ground Truth (GT) LaTeX non-unique, which introduces inherent flaws for the formula evaluation.

As illustrated in Case 1 of Figure 1 earlier, the formula $(x+y)+z = x+(y+z)$ corresponds to the GT annotation `"\left(x+y\right)+z=x+\left(y+z\right)"`. When the model's prediction is `"(x+y)+z=x+(y+z)"`, the prediction is correct because the rendered formula image matches the GT image, despite different LaTeX syntax. Theoretically, the ExpRate/BLEU/Edit Distance results should be 1/1/0, indicating a correct instance. However, in practice, ExpRate is 0, BLEU is 0.449, and Edit Distance is 0.571, failing to accurately assess the formula's quality.

The aforementioned issues make it challenging to objectively evaluate the performance of different formula recognition models. For instance, as illustrated in Case 2 of Figure 1, one character `"z"` is misrecognized as `"2"`. The prediction is incorrect, and the ExpRate, BLEU, and Edit Distance metrics reflect this error. However, when compared to Case 1 where the model prediction is correct, the BLEU and Edit Distance metrics for the incorrect prediction in Case 2 are better than those for the correct prediction in Case 1.

A LaTeX regularization method, which abstracts LaTeX code into a tree structure and standardizes elements, addresses LaTeX syntax diversity [8]. Pix2tex [4], Texify [28], and UniMERNet [30] use such regularization method as a preprocessing step before evaluation, which can solve part of the syntax inconsistency issue. For instance, `"x^b_a"`, `"x^{b}_{a}"`, and `"x_{a}^{b}"` all compile to x_a^b. Directly calculating BLEU scores would not correctly assess the model's prediction quality. Regularized code unifies these into a consistent format, such as always adding curly braces and arranging superscripts before subscripts, contributing to the fairness of subsequent metric calculations. However, regularization does not solve all LaTeX syntax diversity issues. Some symbols have multiple representations, such as `"\leq"` and `"\le"` both representing $\leq$. Exhaustively listing these representations is challenging due to the huge LaTeX symbol library and many additional symbols provided by extension packages (*e.g.*, amsmath, amssymb).

Overall, while regularization mitigates some issues, it does not fully address the inherent limitations of current metrics in evaluating formula recognition performance. This highlights the need for a more robust and comprehensive evaluation metric that can accurately reflect the quality

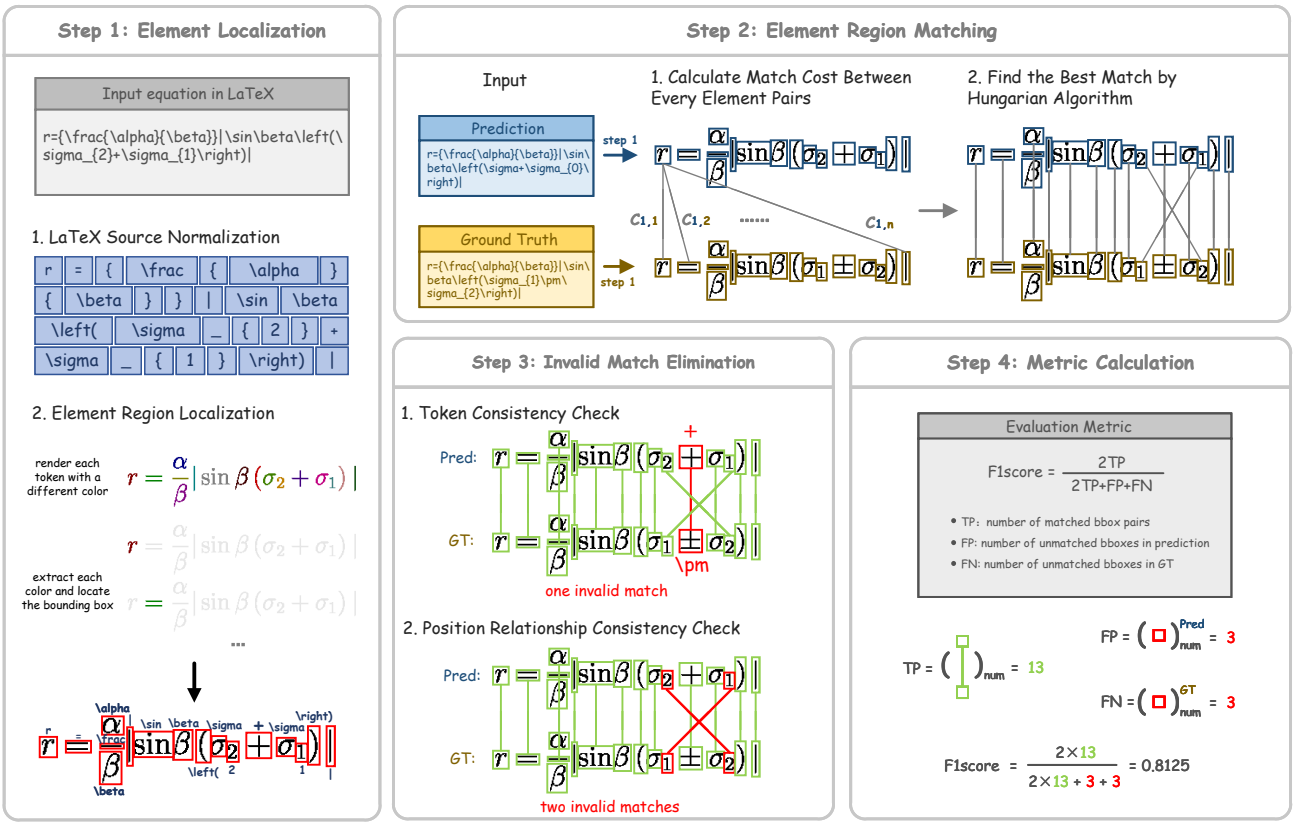


Figure 2. Overview of the Character Detection Matching (CDM), consisting of four main stages. **(1)** Element Localization, where bounding boxes of individual elements are extracted. **(2)** Element Region Matching, which employs a bipartite graph matching method to pair prediction with ground truth elements. **(3)** Invalid Match Elimination, where inconsistent matches are discarded through token and positional relationship checks. **(4)** Metric Calculation, where matching accuracy is evaluated using the F1-Score and ExpRate@CDM.

of formula recognition across diverse representations.

4. Character Detection Matching

Due to the diversity of LaTeX expressions, text-based character-matching methods are unreliable for formula recognition evaluation. The basic idea of CDM is to compare the rendered images from LaTex text. If the image rendered from the predicted LaTeX source code matches the image rendered from the ground truth LaTeX source code, the formula is considered entirely correct. However, directly comparing the pixel values of the original and predicted formulas is not ideal. Any error or extra/missing character in the prediction can cause subsequent characters to be mismatched. Additionally, two similar formulas might have different layouts, with one being a single-line formula and the other a multi-line formula due to line breaks. Therefore, a more robust algorithm is needed to calculate the match between the predicted result and the ground truth image.

To this end, we propose a metric that incorporates a bipartite matching step for element-level matching in images, providing a more intuitive assessment. As shown in Figure 2, the algorithm consists of four stages as follows.

4.1. Element Localization

First, the bounding boxes (bboxes) of each individual element in the rendered image are extracted, followed by the subsequent steps:

LaTeX Source Normalization. LaTeX source codes of both the ground truth and predicted formulas are normalized, breaking them down into individual tokens such as `"2"`, `"a"`, `"A"`, `"\alpha"`, `"\sin"`. Composite elements are decomposed into individual characters, *e.g.*, `"\frac ab"` is decomposed into `"\frac {a} {b}"`.

Element Region Localization. To accurately detect character positions, we render each token in a unique color. We construct an RGB color list with a fixed interval of 15, creating a sequence from $(0, 0, 15)$ to $(255, 255, 255)$. This generates $(255/15 + 1)^3 = 5832$ distinct colors, sufficient for complex formulas. Each token is assigned a unique color using the LaTeX command `"\mathcolor[RGB]{r,g,b}"`. After rendering, we apply image processing techniques to identify pixels match-

ing each color, thereby determining the exact coordinates of each token.

4.2. Element Region Matching

In this stage, a bipartite matching method pairs the predicted elements with the corresponding ground truth elements. Based on the element localization, two sets are obtained for each formula: one for the ground truth independent elements y and one for the predicted independent elements $\hat{y}$. The number of independent elements in each set is N_y and $N_{\hat{y}}$, respectively, with $N = \min(N_y, N_{\hat{y}})$ being the number of elements in the smaller set.

To measure the similarity between y and $\hat{y}$, we match elements in the two sets by identifying the corresponding ground truth element for each predicted element. We use the bipartite matching Hungarian algorithm [13], as described in DETR [6], to find a permutation $\hat{\sigma}$ that minimizes the total matching cost:

$$\hat{\sigma} = \arg\min_{\sigma \in S_N} \sum_{i=1}^{N} L_{\text{match}}(y_i, \hat{y}_{\sigma(i)}), \tag{2}$$

$$L_{\text{match}} = W_t \times L_t + W_p \times L_p + W_o \times L_o, \tag{3}$$

where the matching cost L_{match} is defined as a weighted sum of three components as introduced as follows:

- **Token Matching Cost** L_t: This component measures whether the tokens corresponding to two bounding boxes are the same. If they are identical, the cost is 0; if they are different, the cost is 1. For tokens that render identically but are different, such as `"("`, `"\left("`, and `"\big("`, the cost is 0.05, which can be formulated as follows:

$$L_{\text{t}} = \begin{cases} 0, & \text{if } t_i = \hat{t}_{\sigma(i)}; \\ 0.05, & \text{if } t_i \approx \hat{t}_{\sigma(i)}; \\ 1, & \text{otherwise;} \end{cases} \tag{4}$$

  where $\approx$ denotes tokens that differ but render identically.
- **Positional Proximity Cost** L_p: This component measures the proximity of the two bounding boxes' positions using the L1 norm of their coordinates, which can be formulated as follows:

$$L_{\text{p}} = \frac{1}{D_b} \times \|b_i - \hat{b}_{\sigma(i)}\|_1, \tag{5}$$

  where $b = [x_1, y_1, x_2, y_2]$, and D_b is the dimension of the bounding box coordinates.
- **Order Similarity Cost** L_o: This measures the similarity of the token order in the original LaTeX source (an approximation of reading order). The order is normalized to the range [0, 1], and the L1 norm can be calculated as follows:

$$L_{\text{o}} = \frac{1}{D_o} \times \|o_i - \hat{o}_{\sigma(i)}\|_1, \tag{6}$$

  where o_i means order of i-th token, with dimension D_o.

Overall, the weights W_t, W_p, W_o are used to balance the contributions of the three components. By employing this comprehensive matching strategy, we ensure a more accurate and robust evaluation of the correspondence between the predicted and ground truth elements, thereby improving the overall assessment of formula recognition quality.

4.3. Invalid Match Elimination

After pairing the individual elements of the predicted result with the ground truth using the Hungarian matching algorithm, we need to verify these pairs and eliminate invalid matches. This process involves two steps:

Token Consistency Check. Check whether the elements in each matched pair are consistent in terms of characters. If they are inconsistent, discard the match.

Position Relationship Consistency Check. The relative positions of elements in mathematical formulas are crucial. For instance, in the expressions 2^3 and 3^2, bipartite matching might pair 2 with 2 and 3 with 3, but their meanings and visual representations are entirely different. Thus, we need to check the consistency of the positional relationships within the matched pairs. We treat each element in the matched pair as a bounding box and analyze their relative positions. Specifically, we assume an affine transformation between the ground truth and predicted elements:

$$\hat{b}_{\sigma(i)} = \mathbf{A}(b_i), \tag{7}$$

where $\mathbf{A}$ is the affine transformation matrix. To identify inconsistent match pairs, we detect pairs that do not conform to this transformation relationship. We employ the RANSAC algorithm [9] for this purpose. RANSAC can determine the optimal transformation matrix $\mathbf{A}$ in the presence of noise. Given that formulas are usually horizontally arranged during rendering, we fix the rotation angle in the transformation matrix to 0, considering only translation and scaling. This approach not only improves the convergence speed of the RANSAC algorithm but also enhances the final matching accuracy.

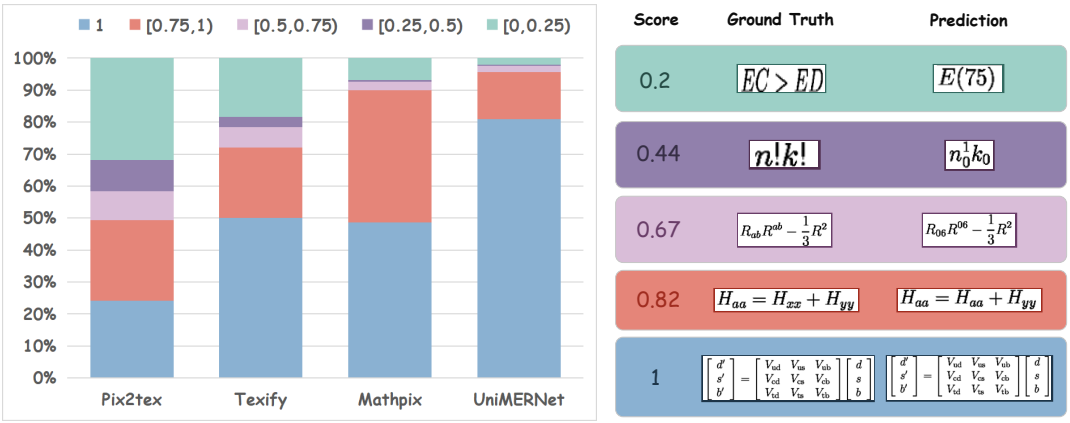
To account for line-breaking effects in formulas, we perform multiple rounds of RANSAC iterations to ensure that as many matched pairs as possible conform to the transformation relationship. After several iterations, matched pairs that still do not conform to the transformation relationship are considered incorrect and are eliminated.

The above two steps effectively eliminate invalid match pairs, ensuring more accurate final matching results.

4.4. Metric Calculation

We use the F1-Score as the default metric for evaluating CDM (Character Detection Metric), defined as:

$$CDM = \frac{2 \times TP}{2 \times TP + FP + FN}, \tag{8}$$



(a) CDM distribution on UniMER-Test dataset

(b) Examples by CDM score range

Figure 3. CDM Score Distribution and Example Analysis. (a) CDM score distribution for Mathpix, UniMERNet, Texify, and Pix2Tex on the UniMER-Test dataset. (b) Example analysis of Pix2Tex predictions across different CDM score ranges.

where TP denotes true positives, FP denotes false positives, and FN denotes false negatives.

To further evaluate the accuracy of formula recognition, we introduce the $ExpRate@CDM$ metric , defined as:

$$ExpRate@CDM = \frac{\sum_{i=1}^{N} \mathbb{I}(CDM_i = 1)}{N}, \quad (9)$$

where $\mathbb{I}$ is the indicator function that equals 1 if $CDM_i = 1$ and 0 otherwise, and N is the total number of formulas. This metric represents the proportion of formulas for which the model's prediction results are perfectly matched. Essentially, $ExpRate@CDM$ serves as a precise version of the ExpRate metric specifically for formula recognition.

5. Experiments

5.1. Models and Data

We validate the CDM metric by evaluating several mainstream formula recognition models using both subjective impressions and objective metrics. The models include open-source UniMERNet [30], Texify [28], Pix2tex [4], and the commercial Mathpix API, all tested on the UniMER-Test dataset. Besides, we evaluate document-level models, such as the open-source Nougat [5] and the commercial GPT-4o [10]. Vary [34] and StrucTexTv3 [21] are excluded as they are currently unavailable.

UniMER-Test Dataset. The dataset includes 23,757 formula samples, categorized into Simple Printed Expressions (SPE), Complex Printed Expressions (CPE), Screenshot Expressions (SCE), and Handwritten Expressions (HWE). We use these categories to conduct the model evaluation.

Tiny-Doc-Math Dataset. To evaluate document-level recognition, we construct the Tiny-Doc-Math dataset, consisting of arXiv papers in mathematics and computer science, published after June 2024, to ensure that they are not in the training data of the compared models. We obtain LaTeX code and corresponding PDFs, match displayed equations using regular expressions, and manually verify them. Overall, the dataset includes 12 PDFs, totaling 196 pages and 437 formulas.

This validation set includes both formula-level and document-level evaluations:

- **Formula-level:** Using single rendered formula images as input, we evaluate Mathpix, Pix2Tex, and UniMERNet. These models accept cropped formula images as input, and we compare the model outputs with the ground truth to compute relevant metrics.
- **Document-level:** Using PDFs or images as input, we evaluate Nougat, GPT-4o, and Mathpix, which can convert entire PDF pages into Markdown format. We match the displayed equations in the model outputs using regular expressions and compare them with the ground truth LaTeX formulas to compute relevant metrics.

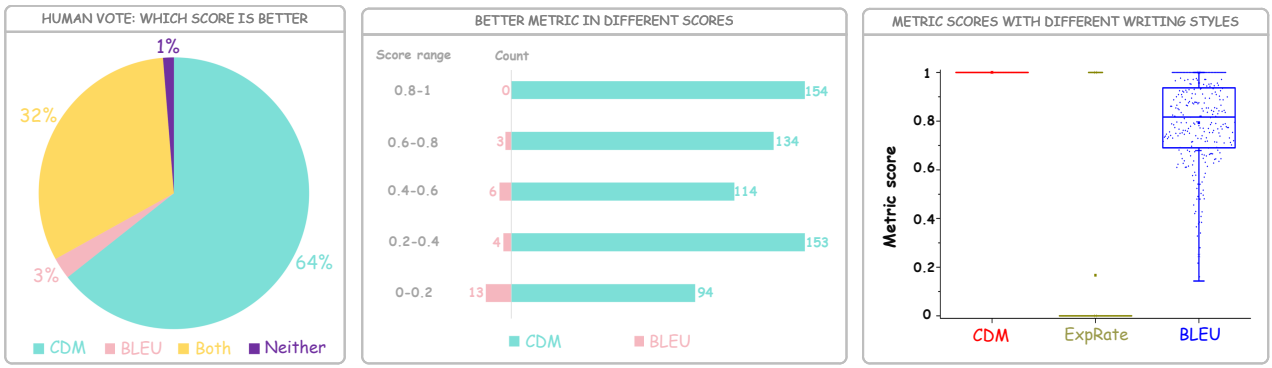
5.2. Credibility Assessment of CDM

5.2.1. Rendering Success Rate

The CDM metric relies on the successful rendering of formula images. For models that fail to render images, we assign a CDM score of 0, as rendering failures indicate that the predicted LaTeX code lacks critical elements. The rendering success rates on the UniMER-Test dataset for Pix2tex, Texify, UniMERNet, and Mathpix are 96.63%, 94.77%, 99.71%, and 97.82%, respectively, ensuring the applicability and reliability of the CDM metric.

5.2.2. User Preference Evaluation

We analyze the distribution of CDM scores for four models on the UniMER-Test dataset. As shown in Figure 3a, Mathpix and UniMERNet perform well in terms of CDM scores. We conduct a detailed analysis of the Pix2Tex model by randomly selecting samples from different score ranges to



(a) Human Evaluation Preferences (b) BLEU vs. CDM Score Distribution (c) Impact of Formula Writing Styles

Figure 4. Human evaluation and metric comparison. (a) Distribution of human preferences. (b) Distribution of cases preferring BLEU or CDM across different score ranges. (c) Impact of formula writing styles on BLEU, ExpRate and CDM scores.

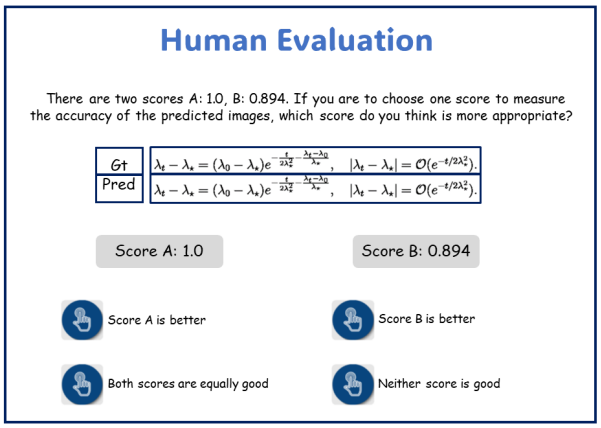


Figure 5. Annotation interface for evaluating user preferences between CDM and BLEU scores. Annotators chose from four options: "Score A is better", "Score B is better", "Both scores are equally good", or "Neither score is good".

evaluate if the prediction quality corresponds to the CDM scores. The analysis in Figure 3b shows that the CDM scores effectively reflect formula quality, with higher scores indicating fewer errors.

To verify the consistency between the CDM metric and human evaluation, we select 1,008 CDM scores from Pix2Tex predictions, ensuring a balanced score distribution. We design an annotation interface (shown in Figure 5), displaying a ground truth label and the corresponding predicted LaTeX rendered image. Annotators choose between ScoreA, ScoreB, Both (credible), and Neither (credible). ScoreA and ScoreB correspond to the BLEU and CDM scores, respectively, but their order is randomized. For more details, please refer to the supplementary materials.

The results in Figure 4a show that 64% of participants prefer the CDM metric, and 32% consider both metrics good. This indicates a 96% consistency between the CDM metric and human evaluation, demonstrating its reliability.

Model	ExpRate	ExpRate@CDM	BLEU	CDM
Pix2tex	0.1237	0.2910	0.4080	0.6360
Texify	0.2288	0.4950	0.5890	0.7550
Mathpix	0.2610	0.5000	0.8067	0.9510
UniMERNet	0.4799	0.8110	0.8425	0.9680

Table 1. UniMER-Test evaluation results. Comparison of mainstream models using different metrics.

Figure 4b compares the number of cases where BLEU or CDM is preferred across different score ranges, showing that CDM consistently outperforms BLEU.

5.2.3. Objective Stability Assessment

To evaluate the impact of formula writing styles on the CDM and BLEU metrics, we randomly select 50 formulas with LaTeX source code and rewrite each formula five times using GPT-4, generating 250 additional formulas. We manually verify these formulas to ensure their rendered results are identical to the original 50 formulas. Using the initial LaTeX source code as the ground truth, we analyze the score distribution of the BLEU and CDM metrics. As shown in Figure 4c, the CDM metric is unaffected by style changes, with all samples scoring 1. In contrast, the BLEU metric's scores are dispersed, making it unsuitable for formula evaluation. The CDM metric remains robust and reliable despite formatting changes.

5.3. Evaluation of Mainstream Models

We conduct a detailed evaluation of mainstream models using both the CDM and BLEU metric. Note that all BLEU metric in this paper have been normalized [4, 8]. However, as discussed in the limitation section, normalization operations cannot address all issues, which will be evident in the following experiments.

5.3.1. UniMER-Test Evaluation

As shown in Table 1, the evaluation results of the four models on the UniMER-Test dataset indicate that the quality

Method	SPE		CPE		HWE		SCE	
	BLEU ↑	CDM ↑	BLEU ↑	CDM ↑	BLEU ↑	CDM ↑	BLEU ↑	CDM ↑
Pix2tex [4]	0.8730	0.9619	0.6550	0.6489	0.0120	0.2453	0.0920	0.6762
Texify [28]	0.9060	0.9852	0.6900	0.7041	0.3410	0.5269	0.4200	0.7932
Mathpix	0.7920	0.9729	0.8061	0.9671	0.8060	0.9318	0.8182	0.9238
UniMERNet [30]	0.9170	0.9946	0.9160	0.9707	0.9210	0.9530	0.6160	0.9461

Table 2. UniMER-Test subset evaluation results. Analysis of anomalies in BLEU and CDM metrics across different subsets.

Image Type	Model	BLEU	CDM	ExpRate@CDM
Formula	Pix2tex	0.4648	0.74440	0.3684
	GPT-4o	0.6431	0.7330	0.4324
	UniMERNet	0.6056	0.9396	**0.6887**
	Mathpix	0.6112	**0.9480**	0.2105
Document	GPT-4o	0.3411	0.6502	0.1670
	Nougat	0.5897	0.8326	0.6086
	Mathpix	0.5939	**0.9567**	**0.6292**

Table 3. Tiny-Doc-Math Evaluation Results. Performance of mainstream models on cropped formula inputs and document-level screenshots using BLEU and CDM metrics.

ranges from low to high as follows: Pix2Tex, Texify, Mathpix, and UniMERNet, based on both BLEU and CDM metrics. ExpRate@CDM clearly shows the proportion of completely correct predictions for each model, indicating that the text character-based ExpRate is unreliable.

From the results in Table 1, it appears that the trends of the BLEU and CDM metrics are consistent. To verify the reliability of using the BLEU metric for model comparison, we further present evaluation results on the UniMER-Test subsets. As shown in Table 2, we observe two notable anomalies: Firstly, in the SCE subset, when comparing the quality of the Mathpix and UniMERNet models, the BLEU and CDM metrics provide opposite conclusions. A detailed review of the UniMERNet paper reveals that the SCE subset was annotated based on Mathpix and then manually corrected. This means that the expression style of the SCE formulas is more consistent with Mathpix. Consequently, even though the CDM metric indicates that UniMERNet has better actual model quality, the BLEU metric, influenced by the expression style, suggests that Mathpix is superior. Secondly, for the Pix2Tex model, the BLEU metric is very low on the HWE and SCE subsets but performs well on the SPE and CPE subsets. This discrepancy arises because the Pix2Tex training set includes a large number of printed formulas from arXiv and lacks data in the HWE and SCE styles.

These anomalies clearly illustrate the limitations of the BLEU metric in evaluating the quality of formula recognition models. In contrast, the CDM metric proposed in this paper is fair and intuitive.

5.3.2. Tiny-Doc-Math Evaluation

The evaluation results of Tiny-Doc-Math are shown in Table 3. For cropped formula inputs (formula-level), all four models perform reasonably well, with CDM scores above 0.7. Notably, the current leading multimodal large model GPT-4o has the highest BLEU score among the four models but the lowest CDM score. This discrepancy indicates that the BLEU metric may not be reliable, suggesting that the formula recognition accuracy of GPT-4o still has room for improvement, lagging behind traditional SOTA models. Additionally, although Mathpix has the highest CDM score, only 21.05% of the formulas are completely accurate. Manual verification revealed that many formulas are missing commas or periods at the end.

When the input is document-level screenshots, the models output the recognition results for the entire document (not just the formulas). Evaluation is conducted by matching the recognized block formulas. In this scenario, it can be observed that the accuracy of GPT-4o further decreases. In contrast, Mathpix and Nougat perform better, but even the document multimodal large model Nougat only achieves a CDM score of 0.8326. This indicates that there is still significant room for improvement in document-level recognition models. Mathpix remains the best performer, with a fully correct formula rate of 62.92%. The accuracy of document-level recognition is crucial for advanced document understanding tasks like scientific knowledge Q&A, and CDM provides an excellent standard for selecting formula models and offers direction for improving formula recognition.

6. Conclusion

In this paper, we introduced Character Detection Matching (CDM), a novel evaluation metric for formula recognition task. CDM addresses the shortcomings of the existing metrics in formula recognition by utilizing spatial character matching, overcoming the challenges posed by diverse formula representations. Comprehensive evaluations on different models and datasets demonstrate CDM's superiority in precisely reflecting recognition quality, showing more intuitive assessment and paving the way for future research and improvements in the formula recognition field.

7. Acknowledgments

This project was supported by National Key R&D Program of China (NO.2022ZD0160102) and Shanghai Artificial Intelligence Laboratory.

References

[1] Francisco Álvaro, Joan-Andreu Sánchez, and José-Miguel Benedí. An integrated grammar-based approach for mathematical expression recognition. *Pattern Recognition*, 51: 135–147, 2016.

[2] Ahmad-Montaser Awal, Harold Mouchere, and Christian Viard-Gaudin. A global learning approach for an online handwritten mathematical expression recognition system. *Pattern Recognition Letters*, 35:68–77, 2014.

[3] Xiaohang Bian, Bo Qin, Xiaozhe Xin, Jianwu Li, Xuefeng Su, and Yanfeng Wang. Handwritten mathematical expression recognition via attention aggregation based bi-directional mutual learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 113–121, 2022.

[4] Lukas Blecher. pix2tex - latex ocr. https://github.com/lukas-blecher/LaTeX-OCR, 2022. Accessed: 2024-2-29.

[5] Lukas Blecher, Guillem Cucurull, Thomas Scialom, and Robert Stojnic. Nougat: Neural optical understanding for academic documents. *arXiv.org*, 2308.13418, 2023.

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision (ECCV)*, pages 213–229. Springer, 2020.

[7] Hiuyi Cheng, Peirong Zhang, Sihang Wu, Jiaxin Zhang, Qiyuan Zhu, Zecheng Xie, Jing Li, Kai Ding, and Lianwen Jin. M6doc: A large-scale multi-format, multi-type, multi-layout, multi-language, multi-annotation category dataset for modern document layout analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15138–15147, 2023.

[8] Yuntian Deng, Anssi Kanervisto, Jeffrey Ling, and Alexander M Rush. Image-to-markup generation with coarse-to-fine attention. In *International Conference on Machine Learning (ICML)*, pages 980–989. PMLR, 2017.

[9] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[10] GPT-4o. Gpt-4o. https://openai.com/index/hello-gpt-4o/, 2024. Accessed: 2024-08-15.

[11] Mingxin Huang, Yuliang Liu, Dingkang Liang, Lianwen Jin, and Xiang Bai. Mini-monkey: Alleviate the sawtooth effect by multi-scale adaptive cropping. *arXiv.org*, 2024.

[12] Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. Ocr-free document understanding transformer. In *European Conference on Computer Vision (ECCV)*, pages 498–517. Springer, 2022.

[13] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[14] Vinay Kukreja et al. Recent trends in mathematical expressions recognition: An lda-based analysis. *Expert Systems with Applications*, 213:119028, 2023.

[15] Stéphane Lavirotte and Loïc Pottier. Mathematical formula recognition using graph grammar. In *Document Recognition V, San Jose, CA, USA, January 24, 1998*, 1998.

[16] Anh Duc Le, Bipin Indurkhya, and Masaki Nakagawa. Pattern generation strategies for improving recognition of handwritten mathematical expressions. *Pattern Recognition Letters*, 128:255–262, 2019.

[17] Vladimir I Levenshtein et al. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, pages 707–710. Soviet Union, 1966.

[18] Bohan Li, Ye Yuan, Dingkang Liang, Xiao Liu, Zhilong Ji, Jinfeng Bai, Wenyu Liu, and Xiang Bai. When counting meets hmer: counting-aware network for handwritten mathematical expression recognition. In *European Conference on Computer Vision (ECCV)*, pages 197–214. Springer, 2022.

[19] Minghao Li, Yiheng Xu, Lei Cui, Shaohan Huang, Furu Wei, Zhoujun Li, and Ming Zhou. Docbank: A benchmark dataset for document layout analysis. *arXiv preprint arXiv:2006.01038*, 2020.

[20] Minghui Liao, Guan Pang, Jing Huang, Tal Hassner, and Xiang Bai. Mask textspotter v3: Segmentation proposal network for robust scene text spotting. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 706–722. Springer, 2020.

[21] Pengyuan Lyu, Yulin Li, Hao Zhou, Weihong Ma, Xingyu Wan, Qunyi Xie, Liang Wu, Chengquan Zhang, Kun Yao, Errui Ding, et al. Structextv3: An efficient vision-language model for text-rich image perception, comprehension, and beyond. *arXiv.org*, 2024.

[22] Scott MacLean and George Labahn. A new approach for recognizing handwritten mathematics using relational grammars and fuzzy sets. *International Journal on Document Analysis and Recognition (IJDAR)*, 16:139–163, 2013.

[23] Mahshad Mahdavi, Richard Zanibbi, Harold Mouchere, Christian Viard-Gaudin, and Utpal Garain. Icdar 2019 crohme+ tfd: Competition on recognition of handwritten mathematical expressions and typeset formula detection. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1533–1538. IEEE, 2019.

[24] Harold Mouchere, Christian Viard-Gaudin, Richard Zanibbi, Utpal Garain, Dae Hwan Kim, and Jin Hyung Kim. Icdar 2013 crohme: Third international competition on recognition of online handwritten mathematical expressions. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1428–1432. IEEE, 2013.

[25] Harold Mouchere, Christian Viard-Gaudin, Richard Zanibbi, and Utpal Garain. Icfhr 2014 competition on recognition of on-line handwritten mathematical expressions (crohme

2014). In *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 791–796. IEEE, 2014.

[26] Harold Mouchère, Christian Viard-Gaudin, Richard Zanibbi, and Utpal Garain. Icfhr2016 crohme: Competition on recognition of online handwritten mathematical expressions. In *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 607–612. IEEE, 2016.

[27] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[28] Vik Paruchuri. Texify. https://github.com/VikParuchuri/texify, 2023. Accessed: 2024-2-29.

[29] Josselin Somerville Roberts, Tony Lee, Chi Heem Wong, Michihiro Yasunaga, Yifan Mai, and Percy Liang. Image2struct: Benchmarking structure extraction for vision-language models. *arXiv preprint arXiv:2410.22456*, 2024.

[30] Bin Wang, Zhuangcheng Gu, Chao Xu, Bo Zhang, Botian Shi, and Conghui He. Unimernet: A universal network for real-world mathematical expression recognition. *arXiv.org*, 2024.

[31] Bin Wang, Chao Xu, Xiaomeng Zhao, Linke Ouyang, Fan Wu, Zhiyuan Zhao, Rui Xu, Kaiwen Liu, Yuan Qu, Fukai Shang, et al. Mineru: An open-source solution for precise document content extraction. *arXiv preprint arXiv:2409.18839*, 2024.

[32] Zelun Wang and Jyh-Charn Liu. Translating math formula images to latex sequences using deep neural networks with sequence-level training. *International Journal on Document Analysis and Recognition (IJDAR)*, 24(1):63–75, 2021.

[33] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004.

[34] Haoran Wei, Lingyu Kong, Jinyue Chen, Liang Zhao, Zheng Ge, Jinrong Yang, Jianjian Sun, Chunrui Han, and Xiangyu Zhang. Vary: Scaling up the vision vocabulary for large vision-language models. *arXiv.org*, 2312.06109, 2023.

[35] Haoran Wei, Lingyu Kong, Jinyue Chen, Liang Zhao, Zheng Ge, Jinrong Yang, Jianjian Sun, Chunrui Han, and Xiangyu Zhang. Vary: Scaling up the vision vocabulary for large vision-language model. In *European Conference on Computer Vision*, pages 408–424. Springer, 2025.

[36] Renqiu Xia, Bo Zhang, Haoyang Peng, Ning Liao, Peng Ye, Botian Shi, Junchi Yan, and Yu Qiao. Structchart: Perception, structuring, reasoning for visual chart understanding. *arXiv preprint arXiv:2309.11268*, 2023.

[37] Renqiu Xia, Song Mao, Xiangchao Yan, Hongbin Zhou, Bo Zhang, Haoyang Peng, Jiahao Pi, Daocheng Fu, Wenjie Wu, Hancheng Ye, et al. Docgenome: An open large-scale scientific document benchmark for training and testing multi-modal large language models. *arXiv preprint arXiv:2406.11633*, 2024.

[38] Renqiu Xia, Bo Zhang, Hancheng Ye, Xiangchao Yan, Qi Liu, Hongbin Zhou, Zijun Chen, Min Dou, Botian Shi, Junchi Yan, et al. Chartx & chartvlm: A versatile benchmark and foundation model for complicated chart reasoning. *arXiv preprint arXiv:2402.12185*, 2024.

[39] Mingkun Yang, Yushuo Guan, Minghui Liao, Xin He, Kaigui Bian, Song Bai, Cong Yao, and Xiang Bai. Symmetry-constrained rectification network for scene text recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9147–9156, 2019.

[40] Ya-Qi Yu, Minghui Liao, Jiwen Zhang, and Jihao Wu. Texthawk2: A large vision-language model excels in bilingual ocr and grounding with 16x fewer tokens. *arXiv preprint arXiv:2410.05261*, 2024.

[41] Ye Yuan, Xiao Liu, Wondimu Dikubab, Hui Liu, Zhilong Ji, Zhongqin Wu, and Xiang Bai. Syntax-aware network for handwritten mathematical expression recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4553–4562, 2022.

[42] Jianshu Zhang, Jun Du, Yongxin Yang, Yi-Zhe Song, Si Wei, and Lirong Dai. A tree-structured decoder for image-to-markup generation. In *International Conference on Machine Learning (ICML)*, pages 11076–11085. PMLR, 2020.

[43] Wenqi Zhao, Liangcai Gao, Zuoyu Yan, Shuai Peng, Lin Du, and Ziyin Zhang. Handwritten mathematical expression recognition with bidirectionally trained transformer. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 570–584. Springer, 2021.