From Pixel to Precision: Enhancing Handwritten Mathematical Expression Recognition with Image-Level Reward

Ze Liu Kai Zhang* Xianquan Wang Shuochen Liu Jiaxian Yan Yupeng Han Qi Liu
University of Science and Technology of China
Hefei, Anhui, China
{liuze2120, wxqcn, shuochenliu, jiaxianyan, yupenghan}@mail.ustc.edu.cn
{kkzhang08, qiliuql}@ustc.edu.cn

Abstract

*Handwritten mathematical expression recognition is hindered by a fundamental misalignment between the dual representations of LaTeX formulas: the symbolic text and the rendered visual image. This discrepancy means that textually distinct LaTeX sequences can produce visually identical outputs, while minor textual errors can cause catastrophic rendering failures. As a result, text-level reward mechanisms cannot perfectly assess the quality of model predictions, failing to effectively guide the model towards optimal performance during training. To overcome this limitation, we introduce the **Image Matching Score (IMS)**, a lightweight yet effective reward based on the structural edit distance of column-wise image projections, which robustly quantifies the visual fidelity between rendered formulas. Leveraging IMS, we then propose **I**mage-**M**atching driven **P**olicy **O**ptimization (**IMPO**), a training framework built upon Group Relative Policy Optimization (**GRPO**). This approach facilitates stable policy learning directly from our sequence-level visual reward, notably without the need for a separate value function network. Extensive experiments demonstrate that IMPO yields consistent performance gains across various backbone models on the challenging CROHME, HME100K, and M^{2}E datasets. Our framework establishes new state-of-the-art results, improving the Expression Recognition Rate by an average of 1.1% and up to 1.37% over strong prior methods.*

1. Introduction

Handwritten Mathematical Expression Recognition (HMER) converts handwritten expressions into LaTeX code and underpins applications in document digitization, office automation, and intelligent education, making accuracy crucial. Although deep neural networks have advanced HMER, it remains difficult due to structural complexity, deformations, diverse writing styles, and 2D symbol arrangements. Existing methods follow two main directions: tree-based models [26, 31, 37], which explicitly parse formula structures but offer limited flexibility, and sequence-based models [2, 4, 40, 41], which treat HMER as an image-to-sequence task and provide greater adaptability.

However, a key limitation of current HMER methods is their reliance on optimizing a *proxy objective*, textual similarity, which often misaligns with the true goal of *visual fidelity*. We define visual fidelity as the **structural and sym-**

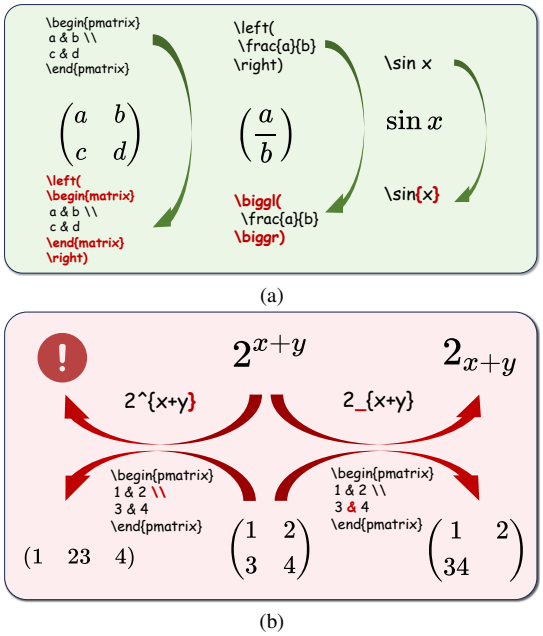


Figure 1. Misalignment between LaTeX syntax and rendered output clearly reveals the inherent and practical limits of text-based metrics. (a) Distinct source codes can yield identical images, causing false negatives. (b) Minor syntax errors may corrupt rendering but receive minimal penalties from text-based scores.

*Corresponding author.

bolic consistency between the rendered formula and the original handwritten input. The main goal of HMER is to generate LaTeX code that accurately reproduces the expression's visual appearance, rather than matching a reference string. Text-level metrics (e.g., BLEU [18], ROUGE [11], exact string match) are inadequate for HMER in two key ways, as shown in Fig. 1. **First**, different LaTeX syntaxes can yield identical renderings (Fig. 1a), causing false negatives. **Second**, small textual errors, such as a missing brace or wrong superscript, can severely distort rendering (Fig. 1b) but incur only mild penalties from character-level metrics, which ignore structural and visual correctness. This issue is pronounced in complex expressions, where reliance on text-based objectives hinders performance.

To address this limitation, we propose **IMPO** (**I**mage-**M**atching driven **P**olicy **O**ptimization), a novel training paradigm that directly optimizes visual fidelity via image-level rewards. Central to our approach is the **I**mage **M**atching **S**core (**IMS**), which quantifies visual similarity between images rendered from predicted and ground-truth LaTeX code. We incorporate IMS into a reinforcement learning framework based on Group Relative Policy Optimization (GRPO), yielding a model-agnostic method fine-tuning sequence-based HMER models for visual correctness instead of textual similarity. Our contributions are as follows:

1) **New Reward:** We propose the **I**mage **M**atching **S**core **(IMS)**, an image-level metric comparing rendered formulas, addressing text-level limitations and capturing expression differences more effectively than standard image metrics.

2) **New Optimization Paradigm:** We present **IMPO**, a model-agnostic reinforcement learning framework that optimizes models using IMS rewards reflecting the visual fidelity of rendered formulas, rather than textual similarity.

3) **Proven Efficacy:** We demonstrate the effectiveness and broad applicability of our image-level optimization paradigm by applying IMPO to several model architectures. IMPO achieves consistent improvements across multiple challenging benchmarks, validating that directly optimizing for visual fidelity is a promising approach for HMER.

2. Related Work

2.1. Handwritten Formula Recognition

The field of Handwritten Mathematical Expression Recognition (HMER) has shifted from traditional grammar-based methods [1, 6, 23, 32] to end-to-end neural models, which fall into two categories, **tree-based** and **sequence-based**.

2.1.1. Tree-based Methods.

Tree-based methods exploit the hierarchical nature of mathematical expressions to parse their structure. Early approaches such as **BLSTM** [38] and **SRD** [36] introduced tree-structured encoding and relationship decoding. Subsequent models, including **TSDNet** [42] and **DenseWAP-TD** [37], extended this paradigm with Transformer architectures, while **TDv2** [26] improved generalization by reducing context dependence. To better capture syntax, **SAN** [31] integrated grammatical constraints, and **BPD** [10] employed parallel branch decoding to shorten generation.

2.1.2. Sequence-based Methods.

The dominant sequence-based paradigm treats HMER as an image-to-sequence task, initiated by **WYGIWYS** [3] and **WAP** [33]. Subsequent improvements included stronger encoders in **DenseWAP** [34] and online trace parsing in **TAP** [35]. Research shifted to enhanced training strategies, such as adversarial learning in **PAL** [27] and **PAL-v2** [28], and auxiliary symbol prediction in **WS-WAP** [22]. Contextual modeling advanced through bidirectional Transformers in **BTTR** [41], mutual learning in **ABM** [2], and coverage-adapted attention in **CoMER** [40], while **CAN** [9] added symbol counting. Recent work targets specific challenges, including multi-line expressions in **LAST** [29] and **GAP** [30]. Other advancements include symbol categorization in **GCN** [39]. The paradigm further diversified with structural priors in **PosFormer** [4], non-autoregressive decoding in **NAMER** [12], implicit character modeling in **ICAL** [43], tree-aware enhancements in **TAMER** [44], and large-scale pre-training in **VLPG** [5].

2.2. Reinforcement Learning

Reinforcement Learning (RL) methods follow two main strategies: value-based and policy-based. Value-based approaches, such as Q-learning [7] and DQN [14], learn value functions to guide decisions, with extensions like Dueling DQN [24] and CQL [8] improving stability. Policy-based methods, including REINFORCE [25] and A3C [15], directly optimize a policy and suit high-dimensional settings. Stability was further improved by TRPO [19] and PPO [20]. Our work builds on Group Relative Policy Optimization (GRPO) [21], a PPO variant that removes the value network by estimating advantages from group-based reward statistics, reducing computation and simplifying training.

3. Method

In this section, we introduce a training framework that addresses a central limitation of existing HMER methods. Conventional approaches, including Maximum Likelihood Estimation (MLE) and reinforcement learning with text-level rewards such as BLEU and ROUGE, emphasize string similarity but cannot guarantee that generated LaTeX renders visually identical to the original expression. We overcome this limitation by shifting the optimization target from text-level to image-level supervision. Specifically, we propose the **I**mage **M**atching **S**core (**IMS**), a fine-grained metric that measures visual similarity between rendered formulas, and use it as a reward within a model-agnostic reinforcement

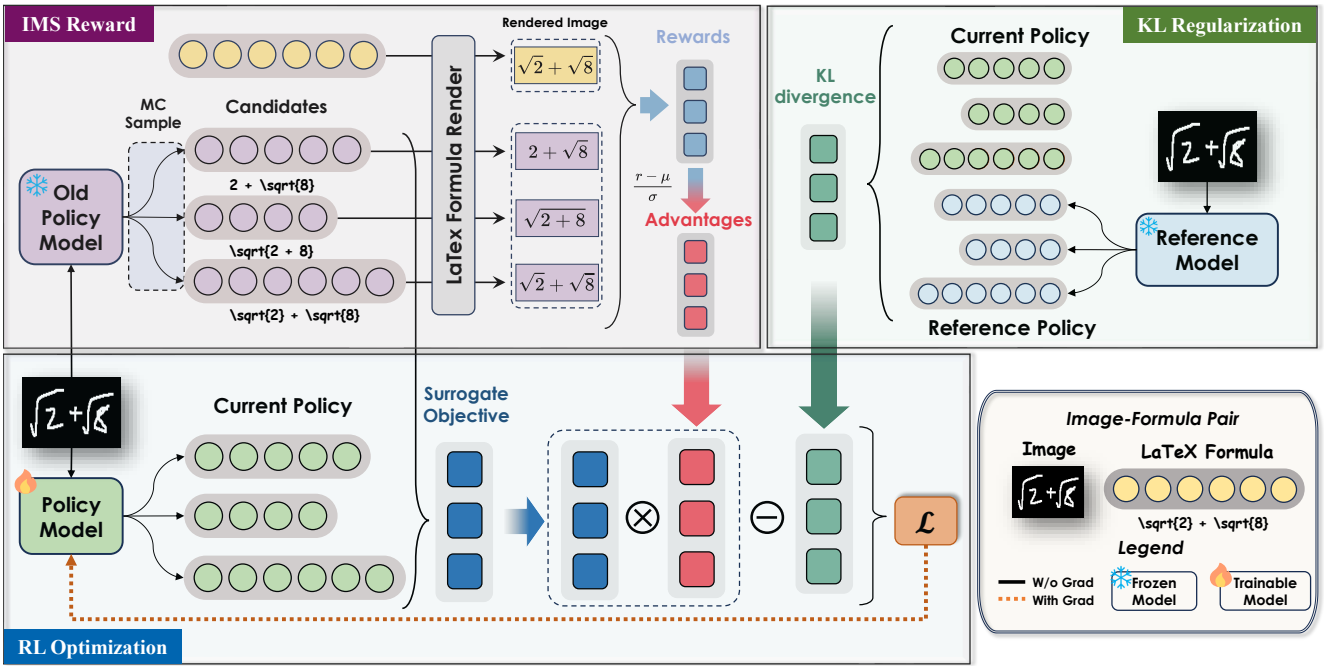


Figure 2. The IMPO framework, built on GRPO, generates image-level rewards. The old policy $\pi_{\theta_{\text{old}}}$ produces candidate sequences, rendered and compared to the target to compute the Image Matching Score (IMS). The normalized IMS yields the advantage, used to update the policy π_θ via a clipped surrogate loss. KL regularization constrains deviations from the $\pi_{\theta_{\text{ref}}}$, enhancing training stability.

learning framework. This enables direct optimization for visual fidelity and improves recognition performance.

3.1. Problem Definition

We formalize the HMER task as follows. Given an input image $X \in \mathbb{R}^{H \times W}$ of a handwritten mathematical expression, the goal is to generate a symbol sequence $Y = \{y^{(t)}\}_{t=1}^{T}$, where each $y^{(t)}$ belongs to a vocabulary $\mathcal{C}$ of LaTeX tokens (e.g., `\frac`, `{`, `}`, `\sin`, `a`, `b`) derived from ground-truth sequences, and T is the sequence length. The challenge is converting visual input into symbolic representations while remaining robust to handwriting and structural variations.

3.2. Image Matching Score

The **I**mage **M**atching **S**core (IMS) is an image-level metric that measures the **visual fidelity** between the rendered image of a predicted LaTeX sequence, F_i, and the ground truth, G_i. The score ranges from $[0, 1]$, with higher values indicating closer alignment. The computation is as follows:

1) Image Preprocessing: To enable consistent comparison, the predicted image F_i and ground-truth image G_i are first padded with white pixels (value 0) to match their maximum height, with content vertically centered. Both images are then binarized using a fixed threshold of 128 to separate foreground symbols from the background. Finally, fully blank columns are removed to eliminate global horizontal shifts. This normalization allows IMS to compare expressions by their *relative layout*, while reducing redundant computation and suppressing background noise for more stable, content-focused evaluation.

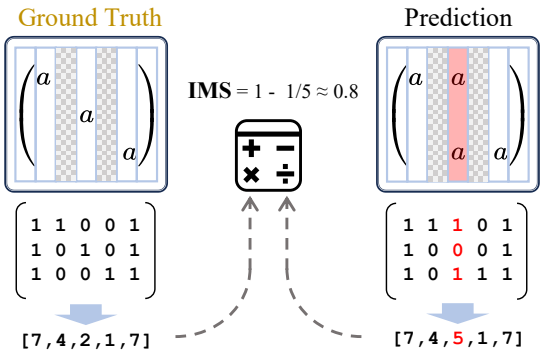


Figure 3. IMS calculation example. A single-character error leads to one mismatch (column '2' becomes '5') in the encoded column sequences. This results in an IMS of $1 - 1/5 = 0.8$.

2) Column-wise Encoding: Each image is transformed into a sequence of integers. For each column, the vertical binary vector is treated as a binary string and converted to its decimal representation. This encoding closely aligns with the spatial layout of mathematical expressions, which are typically arranged from left to right. By dividing the image vertically into columns, IMS effectively captures horizontal compositional relationships while preserving fine-

grained local vertical patterns. It is highly sensitive to local structural errors such as misplaced superscripts or visually similar character substitutions (e.g., "V" and "v").

3) Similarity Calculation: Similarity is computed using the standard Levenshtein edit distance D_i between the two integer sequences $S_P = \{s_P^{(1)}, \ldots, s_P^{(W_P)}\}$ and $S_G = \{s_G^{(1)}, \ldots, s_G^{(W_G)}\}$. The distance is normalized by the length of the longer sequence, $L_i = \max(|S_P|, |S_G|)$, yielding the final similarity score defined as:

$$\text{IMS}(F_i, G_i) = 1 - \frac{D_i}{L_i}, \quad \text{IMS} \in [0, 1]. \tag{1}$$

3.3. IMPO Framework

IMPO reframes HMER as a reinforcement learning problem in which the policy network is rewarded for producing *visually faithful* LaTeX expressions, and builds on Group Relative Policy Optimization (GRPO) [21]. GRPO streamlines training and reduces computation by estimating advantages through reward normalization within sampled groups. This makes it well suited for image-level IMS rewards, enabling stable policy learning without a value network. IMPO optimizes the policy through advantage estimation and updates driven by a tailored loss function.

Advantage Estimation. Given an input image, we sample a batch of N candidate LaTeX sequences using the current policy $\pi_{\theta_{\text{old}}}$. For each candidate Y_i, we compute a sequence-level reward R_i using the IMS score against ground truth. To stabilize training, rewards are normalized across the batch, yielding advantage estimate $\hat{A}_i$, constant across all timesteps t in trajectory i. This provides an effective credit assignment strategy for the entire expression.

Loss Function. The policy is optimized by minimizing a composite objective that balances the clipped surrogate loss with a KL regularization term. Specifically, for each timestep t in trajectory i, the clipped surrogate objective $\mathcal{L}_{\text{CLIP}}$ is formulated as:

$$\mathcal{L}_{\text{CLIP}}(\theta) = \min\Big(\rho_{i,t}\hat{A}_i,\ \text{clip}(\rho_{i,t}, 1 \pm \epsilon)\hat{A}_i\Big), \tag{2}$$

where $\rho_{i,t} = \frac{\pi_\theta(y_{i,t}|y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t}|y_{i,<t})}$ denotes the importance ratio.

To maintain training stability, we define the step-wise joint loss $\mathcal{L}_{i,t}$ by incorporating the KL divergence from a reference policy $\pi_{\theta_{\text{ref}}}$:

$$\mathcal{L}_{i,t}(\theta) = -\mathcal{L}_{\text{CLIP}}(\theta) + \beta\, D_{\text{KL}}(\pi_\theta \parallel \pi_{\theta_{\text{ref}}}). \tag{3}$$

The final policy loss $\mathcal{L}(\theta)$ is then computed as the expectation over the sampled batch of trajectories τ_i and their respective timesteps:

$$\mathcal{L}(\theta) = \mathbb{E}_{\tau_i \sim \pi_{\theta_{\text{old}}}}\left[\sum\nolimits_t \mathcal{L}_{i,t}(\theta)\right]. \tag{4}$$

4. Experiment

In this section, we present comprehensive experiments to thoroughly evaluate the effectiveness of our proposed **IMPO** framework. Our evaluation is carefully designed to address the following key research questions:
1) RQ1: Does IMPO achieve state-of-the-art performance on standard HMER benchmarks?
2) RQ2: What advantages does the proposed image-level reward (IMS) offer over conventional text-level objectives?
3) RQ3: How effectively does the IMPO framework generalize across different model architectures?
4) RQ4: How does IMS compare to other image-level metrics (e.g., SSIM) in robustness and effectiveness as reward?

4.1. Experimental Setup

4.1.1. Datasets and Evaluation Metrics.

We evaluate IMPO on three established benchmarks with diverse challenges. A brief overview is given below:

- **CROHME (2014, 2016, 2019)** [13, 16, 17]: The primary benchmark for single-line HMER, comprising 8,836 training images and three distinct test sets.
- **HME100K** [31]: A large-scale dataset with 74,502 training and 24,607 test images, featuring real-world writing styles that challenge model robustness.
- **M^{2}E** [29]: A multi-line expression benchmark with 79,979 training and 9,985 test images for evaluating complex 2D spatial structure handling.

Following standard HMER practice, we use **Expression Recognition Rate (ExpRate)** as the primary evaluation metric. ExpRate denotes the percentage of expressions whose predicted LaTeX matches the ground truth, serving as a strict correctness criterion. To provide a broader performance view, we also report relaxed metrics allowing up to one or two errors, denoted as "≤ 1 error" and "≤ 2 errors".

4.1.2. Implementation Details.

For experiments, we apply IMPO to the state-of-the-art **PosFormer** [4] to showcase its potential. To evaluate generalizability, we test it on baselines in ablation studies, including **CoMER** [40], **BTTR** [41], and **ABM** [2]. Experiments used NVIDIA RTX 4090 D GPUs. For IMPO, we set the group size to 16, the PPO clipping $\epsilon = 0.2$ and KL coefficient $\beta = 0.01$. A cosine learning rate scheduler is used with 10% warm-up and a base learning rate of 1.5×10^{-5}.

4.2. Comparison with SOTA Methods (RQ1)

We first compare the performance of IMPO against existing state-of-the-art methods on the three benchmark datasets.

Results on CROHME Benchmarks: We evaluate IMPO on CROHME 2014, 2016, and 2019, standard benchmarks for handwritten mathematical expression recognition. As

Table 1. Performance of IMPO on the CROHME2014, 2016, and 2019 datasets, using PosFormer as the base model. "**Bold**" entries indicate state-of-the-art results, while "underlined" entries denote sub-optimal performance. The models are categorized into three groups, listed from top to bottom: RNN-based models, tree-based models, and transformer-based models.

Method	CROHME 2014			CROHME 2016			CROHME 2019		
	ExpRate↑	≤1↑	≤2↑	ExpRate↑	≤1↑	≤2↑	ExpRate↑	≤1↑	≤2↑
WAP [33]	46.55	61.16	65.21	44.55	57.10	61.55	-	-	-
TAP [35]	48.47	63.28	67.34	44.81	59.72	62.77	-	-	-
PAL [27]	39.66	56.80	65.11	-	-	-	-	-	-
PAL-v2 [28]	48.88	64.50	69.78	49.61	64.08	70.27	-	-	-
DenseWAP [34]	50.1	-	-	47.5	-	-	-	-	-
DenseWAP-MSA [34]	52.8	68.1	72.0	50.1	63.8	67.4	47.7	59.5	63.3
WS-WAP [22]	53.65	-	-	51.96	64.34	70.10	-	-	-
ABM [2]	56.85	73.73	81.24	52.92	69.66	78.73	53.96	71.06	78.65
CAN-DWAP [9]	57.00	74.21	80.61	56.06	71.49	79.51	54.88	71.98	79.40
CAN-ABM [9]	57.26	74.52	82.03	56.15	72.71	80.30	55.96	72.73	80.57
DenseWAP-TD [37]	49.1	64.2	67.8	48.5	62.3	65.3	51.4	66.1	69.1
TDv2 [26]	53.62	-	-	55.18	-	-	58.72	-	-
SAN [31]	56.2	72.6	79.2	53.6	69.6	76.8	53.5	69.3	70.1
BTTR [41]	53.96	66.02	70.28	52.31	63.90	68.61	52.96	65.97	69.14
GCN [39]	60.00	-	-	58.94	-	-	61.63	-	-
BPD-Coverage [10]	60.65	-	-	58.50	-	-	61.47	-	-
ICAL [43]	60.63	75.99	82.80	58.79	76.06	83.38	60.51	78.00	84.63
NAMER [12]	60.51	75.03	82.25	60.24	73.50	80.21	61.72	75.31	82.07
VLPG [5]	60.41	74.14	78.90	60.51	75.50	79.86	62.34	76.81	81.15
CoMER [40]	59.33	71.70	75.66	59.81	74.37	80.30	62.97	77.40	81.40
TAMER [44]	61.23	76.77	83.25	60.26	76.91	84.05	61.97	78.97	85.80
PosFormer [4]	62.68	79.01	84.69	61.03	77.86	84.74	64.97	82.49	87.24
IMPO	**63.89**$_{+1.21}$	**80.84**$_{+1.83}$	**88.18**$_{+3.49}$	**62.04**$_{+1.01}$	**80.38**$_{+2.52}$	**88.38**$_{+3.64}$	**66.34**$_{+1.37}$	**85.52**$_{+3.03}$	**89.72**$_{+2.48}$

Table 2. Performance on HME100K and M^2E.

Dataset	Method	ExpRate↑	≤1↑	≤2↑
HME100K	DenseWAP	61.85	70.63	77.14
	ABM	65.93	81.16	87.86
	SAN	67.10	-	-
	CAN-DWAP	67.31	82.93	89.17
	CAN-ABM	68.09	83.22	89.91
	BTTR	64.10	-	-
	CoMER	68.12	84.20	89.71
	ICAL	69.06	85.16	90.61
	NAMER	68.52	83.10	89.30
	TAMER	69.50	85.48	90.80
	PosFormer	69.51	84.91	90.51
	IMPO	**70.67**$_{+1.16}$	**88.23**$_{+2.75}$	**93.54**$_{+2.93}$
M^2E	DenseWAP	53.14	70.15	78.34
	WS-WAP	55.20	72.13	80.13
	ABM	55.48	72.05	80.06
	BTTR	55.25	72.09	80.17
	CoMER	56.20	73.39	81.11
	LAST [29]	56.50	72.80	80.81
	PosFormer	58.33	75.58	83.22
	IMPO	**59.56**$_{+1.23}$	**78.22**$_{+2.64}$	**86.39**$_{+3.17}$

shown in Tab. 1, IMPO consistently achieves state-of-the-art performance. On CROHME 2019, it attains an ExpRate of **66.34%**, exceeding PosFormer (64.97%) by **1.37%**. Under relaxed evaluation, it improves "$\leq$ 1 error" accuracy by 3.03 points (from 82.49% to 85.52%), reducing both the frequency and severity of errors. Gains on CROHME 2014 (+1.21%) and 2016 (+1.01%) further demonstrate IMPO's robustness to diverse symbol sets and handwriting styles.

Results on HME100K and M^2E: Robust recognition in unconstrained environments is vital for practical deployment. To evaluate IMPO's generalization, we test it on HME100K and M^2E, two real-world datasets with diverse styles and noise. As shown in Tab. 2, IMPO achieves an ExpRate of **70.67%**, surpassing PosFormer by 1.16%, with larger gains under fault-tolerant metrics. This confirms that our image-level reward improves resilience to input variability. On M^2E, which demands complex 2D layout understanding, IMPO attains **59.56%** ExpRate, outperforming prior methods by 1.23%. Consistent gains in relaxed accuracy underscore IMPO's advantage in structural parsing, a known limitation of text-level objectives.

These results demonstrate IMPO's effectiveness across both standard benchmarks and real-world datasets, with consistent gains in ExpRate and fault-tolerant metrics validating the image-level reward approach.

4.3. Ablation Analysis

We conduct ablation studies on all three datasets to assess the contribution of each IMPO component and identify the sources of its performance gains.

4.3.1. Effect on Formula Complexity (RQ2)

To assess the benefit of the IMS reward, we compare it with text-level rewards (BLEU-4, ROUGE-L) and the vanilla CoMER and BTTR baselines under the GRPO framework, stratifying CROHME 2014 and 2016 test sets by formula length. Figure 4 reports ExpRate for short ($[0, 20]$),

Table 3. Performance comparison of different models (ABM, CoMER, PosFormer) under Vanilla and +IMPO settings across datasets. "*N/A*" indicates that the corresponding experiment was not conducted.

Dataset	Method	ABM			CoMER			PosFormer		
		ExpRate↑	≤1↑	≤2↑	ExpRate↑	≤1↑	≤2↑	ExpRate↑	≤1↑	≤2↑
CROHME 2014	Vanilla	56.85	73.73	81.24	59.33	71.70	75.66	62.68	79.01	84.69
	+IMPO	$57.98_{+1.13}$	$75.89_{+2.16}$	$83.61_{+2.37}$	$60.72_{+1.39}$	$76.17_{+4.47}$	$83.21_{+7.55}$	$63.89_{+1.21}$	$80.84_{+1.83}$	$88.18_{+3.49}$
CROHME 2016	Vanilla	52.92	69.66	78.73	59.81	74.37	80.30	61.03	77.86	84.74
	+IMPO	$53.78_{+0.86}$	$72.49_{+2.83}$	$79.06_{+0.33}$	$60.97_{+1.16}$	$77.69_{+3.32}$	$85.33_{+5.03}$	$62.04_{+1.01}$	$80.38_{+2.52}$	$88.38_{+3.64}$
CROHME 2019	Vanilla	53.96	71.06	78.65	62.97	77.40	81.40	64.97	82.49	87.24
	+IMPO	$54.64_{+0.68}$	$72.49_{+1.43}$	$80.26_{+1.61}$	$63.98_{+1.01}$	$80.12_{+2.72}$	$85.65_{+4.25}$	$66.34_{+1.37}$	$85.52_{+3.03}$	$89.72_{+2.48}$
HME100K	Vanilla	65.93	81.16	87.86	68.12	84.20	89.71	69.51	84.91	90.51
	+IMPO	*N/A*	*N/A*	*N/A*	$69.33_{+1.21}$	$85.48_{+1.28}$	$91.18_{+1.47}$	$70.67_{+1.16}$	$88.23_{+3.32}$	$93.54_{+3.03}$
M^2E	Vanilla	55.48	72.05	80.06	56.20	73.39	81.11	58.33	75.58	83.22
	+IMPO	*N/A*	*N/A*	*N/A*	$57.76_{+1.56}$	$75.21_{+1.82}$	$81.87_{+0.76}$	$59.56_{+1.23}$	$78.22_{+2.64}$	$86.39_{+3.17}$

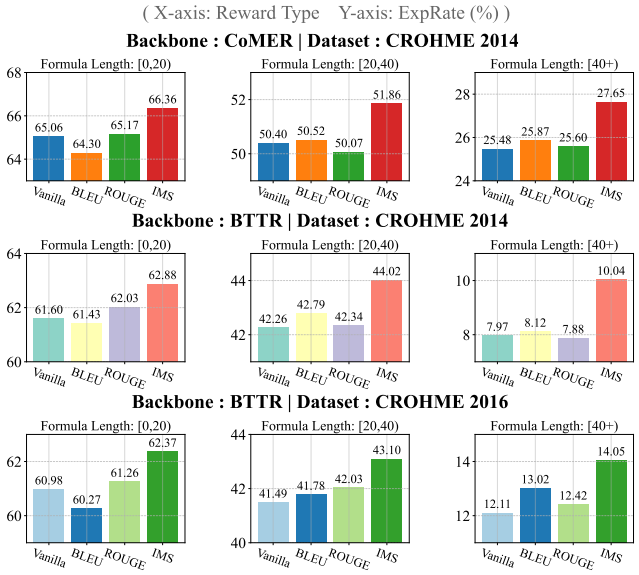


Figure 4. Accuracy comparison between CoMER, BTTR, and reward-based variants on the CROHME 2014 and CROHME 2016 datasets across different formula lengths.

medium ($[20, 40]$), and long ($40+$) formulas. In the CoMER–CROHME 2014 setting, BLEU-4 and ROUGE-L yield minimal gains, whereas IMS consistently outperforms all baselines across lengths, reaching **66.36%** on short and **27.65%** on long formulas, with improvements 1.3% and 2.17%. This pattern holds across backbones (BTTR) and datasets (CROHME 2014 and 2016), demonstrating the robustness of IMS. Overall, the results show that text-level rewards fail to capture structural correctness as complexity grows, while IMS effectively promotes global layout fidelity and accurate modeling of complex expressions.

4.3.2. Generalization Across Architectures (RQ3)

A key advantage of our framework is its model-agnostic design. To address RQ3, we evaluate IMPO across ABM, CoMER, and PosFormer. As shown in Tab. 3, IMPO consistently delivers ExpRate gains across models and datasets (e.g., ABM: 1.13%; CoMER: 1.39%). This generality holds in real-world settings, with similar improvements on HME100K and M^2E. Consistency across different architectures indicates that IMPO addresses optimization misalignment rather than architecture-specific properties, confirming its broad applicability as an effective enhancement for sequence-based HMER systems.

4.3.3. Ablation of Key Components (RQ4)

We justify our design choices through ablation studies on the CoMER backbone across the CROHME datasets, with results shown in Table 5. We compare different RL frameworks and reward variants, where **REIN-b** denotes REINFORCE with baseline, $\mathbf{IMS_{HDiv}}$ applies horizontal image division, and $\mathbf{IMS_{KBC}}$ retains blank columns.

RL Framework Ablation: We compare optimization strategies to assess the general applicability of the IMS reward. As shown in Table 5, integrating IMS yields consistent gains across RL frameworks. With REINFORCE (IMS & REIN-b), the ExpRate on CROHME 2014 increases from 59.33 to 59.77, and with GRPO (IMS & GRPO), it rises to 60.72. All frameworks exhibit clear improvements, demonstrating that the image-level reward enhances performance across standard policy optimization algorithms.

IMS vs. $\mathbf{IMS_{KBC}}$ and $\mathbf{IMS_{HDiv}}$: We analyze the IMS design by comparing it with its variants in Table 5. The $\mathrm{IMS_{HDiv}}$ variant yields only a marginal gain (59.68 on CROHME 2014) relative to standard IMS (60.72), confirming the importance of column-wise processing over horizontal slicing. The $\mathrm{IMS_{KBC}}$ variant performs similarly (60.36), indicating that removing blank columns provides a small but consistent advantage. Penalizing horizontal shifts yields limited gains, as such misalignments are rare with our fixed renderer.

Table 4. Analysis of the robustness of IMS and SSIM to DPI changes under different formula lengths.

Metric	Coefficient of Variation (%) across Length Bins						
	all	**(0, 10]**	**(10, 20]**	**(20, 30]**	**(30, 40]**	**(40, 50]**	**50+**
IMS	**0.62**	**0.61**	**0.89**	**0.18**	**0.78**	**0.48**	**1.34**
SSIM	1.21	0.64	1.02	1.82	2.71	5.32	7.03

Table 5. Performance of different training methods on the CoMER backbone across CROHME datasets. IMS reward variants include **HDiv** (Horizontal Division) and **KBC** (Keep Blank Columns). The RL method variant **REIN-b** denotes REINFORCE with baseline. "*N/A*" in the Reward and RL Method columns indicates that no reward or RL method is used in the vanilla setting.

Dataset	Reward	RL Method	ExpRate↑	≤1↑	≤2↑
CROHME 2014	*N/A*	*N/A*	59.33	71.70	75.66
	SSIM	GRPO	60.11	73.84	79.13
	IMS	REIN-b	59.77	73.53	78.11
	IMS_{HDiv}	GRPO	59.68	73.22	77.84
	IMS_{KBC}	GRPO	60.36	75.85	81.78
	IMS	GRPO	**60.72**	**76.17**	**83.21**
CROHME 2016	*N/A*	*N/A*	59.81	74.37	80.30
	SSIM	GRPO	60.44	76.91	83.16
	IMS	REIN-b	60.34	76.39	83.08
	IMS_{HDiv}	GRPO	60.28	76.18	82.83
	IMS_{KBC}	GRPO	60.59	77.26	84.36
	IMS	GRPO	**60.97**	**77.69**	**85.33**
CROHME 2019	*N/A*	*N/A*	62.97	77.40	81.40
	SSIM	GRPO	63.69	79.43	84.02
	IMS	REIN-b	63.55	79.36	83.97
	IMS_{HDiv}	GRPO	63.39	78.95	83.75
	IMS_{KBC}	GRPO	63.80	79.81	84.98
	IMS	GRPO	**63.98**	**80.12**	**85.65**

IMS vs. SSIM: To evaluate stability of IMS and SSIM under varying image clarity, we partition the CROHME 2014 test set by formula length (step size 10). For each group, we render ground-truth formulas and CoMER predictions at a fixed font size of 10, increasing DPI from 150 to 300 in increments of 25. We compute IMS and SSIM across settings and report their Coefficients of Variation. As shown in Tab. 4, IMS is more robust to resolution changes, with a variation of **0.62%** versus **1.21%** for SSIM. While both metrics behave similarly for short formulas ([0, 10]), SSIM becomes increasingly unstable with length (up to 7.03% for 50+), whereas IMS remains stable. This robustness improves downstream performance: in Table 5, IMS as a GRPO reward consistently outperforms SSIM. On CROHME 2014, IMS & GRPO achieves **60.72** vs. **60.11** for SSIM & GRPO, with similar gains on CROHME 2016 (60.97 vs. 60.44) and CROHME 2019 (63.98 vs. 63.69). Results show IMS is more stable under resolution shifts and more effective for optimizing visual fidelity.

4.4. Case Study: IMPO Framework Performance

To qualitatively assess IMPO, we conduct case studies on three benchmarks: **CROHME2014**, **HME100K**, and **M^{2}E**,

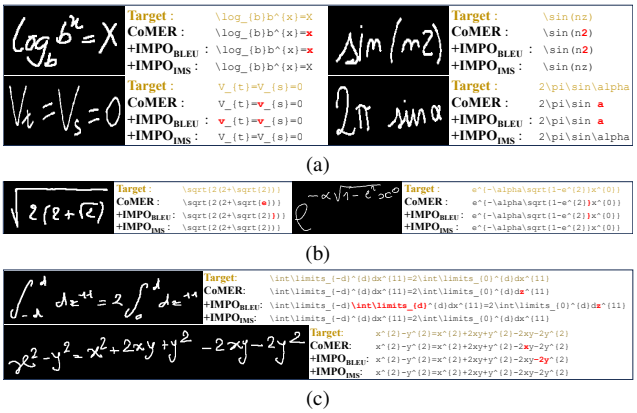


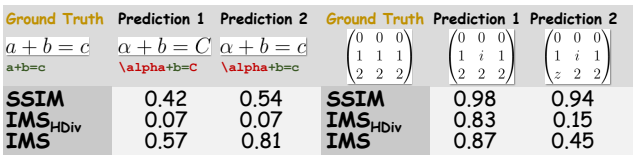
Figure 5. Case studies on CROHME 2014 using CoMER. (a) IMPO improves discrimination of visually similar characters. (b) IMPO preserves delimiters, enhancing syntactic correctness. (c) IMPO increases robustness on structurally complex expressions.

comparing the baseline CoMER with its IMPO-enhanced variants using BLEU or IMS as rewards. The analysis targets three main challenges: (i) symbol-level ambiguities, (ii) structural consistency, and (iii) accuracy of operators/relations and expression completeness.

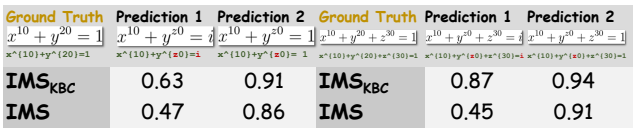
(a) Sensitivity to Visually Similar Characters. Figure 5a highlights the limitations of CoMER and $\text{IMPO}_{\text{BLEU}}$ in distinguishing visually similar characters such as X, z, V, and α. These models frequently confuse these characters with similar-looking ones, producing substitutions like X to x, z to 2, V to v, and α to a. These substitutions lead to semantic misinterpretations in mathematical contexts. IMPO_{IMS} mitigates this by leveraging pixel-level alignment and structural visual cues, enabling better symbol disambiguation and more accurate predictions.

(b) Syntax Preservation via Delimiter Recognition. Figure 5b demonstrates failure cases involving delimiters such as braces and brackets. Both CoMER and $\text{IMPO}_{\text{BLEU}}$ frequently omit critical syntax elements like "{" and "}", which are essential for semantic correctness and visual layout in LaTeX-rendered formulas. In contrast, IMPO_{IMS} benefits from a built-in syntax consistency signal within IMS, enabling the model to recognize and correctly place these elements. As a result, the outputs retain grammatical validity and layout fidelity.

(c) Structural Robustness on Complex Expressions. Figure 5c presents complex multi-component expressions that challenge sequence modeling and layout alignment. CoMER and $\text{IMPO}_{\text{BLEU}}$ often fail to capture long-range dependencies or nested structure, leading to incomplete or misaligned predictions. IMPO_{IMS} consistently produces



(a)



(b)

Figure 6. Case studies illustrating the limitations of SSIM and IMS variants (IMS_{HDiv}, IMS_{KBC}). (a) Comparison with SSIM and IMS_{HDiv}, showing SSIM's weak semantic discrimination and a blind spot in IMS_{HDiv} when handling horizontal structures. (b) Comparison with IMS_{KBC}, showing that retaining blank columns reduces error sensitivity in longer expressions.

accurate representations by evaluating structural integrity through rendered visual forms, demonstrating its robustness in complex recognition scenarios.

4.5. Case Study: IMS Metric Design

To justify our design choices for IMS, we conduct qualitative case studies in Fig. 6, comparing our method against SSIM and two key variants (IMS_{HDiv} and IMS_{KBC}). All LaTeX expressions for these comparisons are rendered using 'KaTeX' with the default 12pt Computer Modern font. The output images are generated at 200 DPI and tightly cropped to remove surrounding white padding.

Limitations of SSIM and IMS_{HDiv}: Figure 6a illustrates limitations of SSIM. Designed for natural images, SSIM emphasizes luminance and texture over symbolic semantics, making it ineffective at distinguishing visually similar but semantically different symbols. In the horizontal example, SSIM fails to differentiate substitutions between 'a' and 'α', and between 'c' and 'C'. Prediction 2, with one error (substituting 'a' with 'α'), scores 0.54, whereas Prediction 1, which also replaces 'c' with 'C', scores 0.42, showing only a small decrease despite an additional error. In the matrix example, Prediction 1 (replacing '1' with 'i') scores 0.98, and Prediction 2 (further substituting '2' with 'z') still scores 0.94. These small differences (0.54 vs. 0.42, 0.98 vs. 0.94) indicate that SSIM does not adequately penalize semantic errors between visually similar symbols because it relies heavily on local texture correlations, which often remain highly consistent even when the underlying symbolic intent differs substantially.

The case study highlights the weakness of IMS_{HDiv}, which partitions images horizontally. As mathematical formulas are mainly arranged in this direction, small symbol errors can affect many partitions, creating an evaluation blind spot. In the first example, IMS_{HDiv} assigns the same score (0.07) to "$\alpha + b = C$" and "$\alpha + b = c$", despite differing error severities. For vertically structured expressions like matrices, both IMS and IMS_{HDiv} differentiate errors effectively (e.g., IMS_{HDiv}: 0.83 vs. 0.15; IMS: 0.87 vs. 0.45). This shows that IMS is robust across layouts, while IMS_{HDiv} struggles with horizontal arrangements. By encoding structural alignment, IMS produces scores that better reflect error severity.

Limitations of IMS_{KBC}: To analyze the behavior of IMS variants, Fig. 6b presents examples illustrating the limitations of IMS_{KBC}, which retains all blank columns during comparison. In Example 1, a short expression, both IMS_{KBC} and IMS distinguish the severity between Prediction 1 (two errors, "$y^{z0} = i$") and Prediction 2 (one error, "$y^{z0} = 1$"), with score gaps of 0.28 ($0.91 - 0.63$) and 0.39 ($0.86 - 0.47$). In Example 2, where the same errors occur in a longer expression, many blank columns significantly inflate the sequence length used for normalization. This reduces the sensitivity of IMS_{KBC}, compressing the score difference to 0.07 ($0.94 - 0.87$). In contrast, IMS, which removes blank columns, maintains a clear margin of 0.46 ($0.91 - 0.45$). These results show that IMS_{KBC} becomes less sensitive as formula length increases and blank regions dominate the comparison.

A second limitation of IMS_{KBC} is its susceptibility to evaluation bias from horizontal shifts. Because blank columns are preserved, IMS_{KBC} would incorrectly penalize two identical formulas if one is rendered with a horizontal shift (e.g., comparing "$E = mc^2$" and "$E \quad = mc^2$"). This misalignment would result in a score less than 1.0, whereas IMS, by removing leading/trailing blank columns, remains invariant and correctly scores 1.0. Although this specific scenario will not occur during training due to fixed rendering settings, it highlights an inherent weakness of IMS_{KBC}.

5. Conclusion

In this paper, we propose **I**mage-**M**atching driven **P**olicy **O**ptimization (**IMPO**) for Handwritten Mathematical Expression Recognition (HMER), which optimizes alignment between predicted and target rendered formula images via the **I**mage **M**atching **S**core (IMS). Our method improves recognition, especially for complex structures. Experiments on CROHME, HME100K, and M^2E show consistent gains in accuracy and fault tolerance, surpassing state-of-the-art methods. Powered by IMS, IMPO provides a flexible, model-agnostic training framework for sequence-based architectures, advancing HMER and guiding future research.

Acknowledgments

This research was partially supported by the National Natural Science Foundation of China (Grants No.62406303) and Anhui Province Science and Technology Innovation Project (202423k09020010).

References

[1] Francisco Álvaro, Joan-Andreu Sánchez, and José-Miguel Benedí. Recognition of on-line handwritten mathematical expressions using 2d stochastic context-free grammars and hidden markov models. *Pattern Recognition Letters*, 35:58–67, 2014. 2

[2] 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*, pages 113–121, 2022. 1, 2, 4, 5

[3] 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*, pages 980–989. PMLR, 2017. 2

[4] Tongkun Guan, Chengyu Lin, Wei Shen, and Xiaokang Yang. Posformer: recognizing complex handwritten mathematical expression with position forest transformer. In *European Conference on Computer Vision*, pages 130–147. Springer, 2024. 1, 2, 4, 5

[5] Hong-Yu Guo, Chuang Wang, Fei Yin, Xiao-Hui Li, and Cheng-Lin Liu. Vision–language pre-training for graph-based handwritten mathematical expression recognition. *Pattern Recognition*, 162:111346, 2025. 2, 5

[6] Jaekyu Ha, Robert M Haralick, and Ihsin T Phillips. Understanding mathematical expressions from document images. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, pages 956–959. IEEE, 1995. 2

[7] Hado Hasselt. Double q-learning. *Advances in neural information processing systems*, 23, 2010. 2

[8] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in neural information processing systems*, 33:1179–1191, 2020. 2

[9] 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*, pages 197–214. Springer, 2022. 2, 5

[10] Zhe Li, Wentao Yang, Hengnian Qi, Lianwen Jin, Yichao Huang, and Kai Ding. A tree-based model with branch parallel decoding for handwritten mathematical expression recognition. *Pattern Recognition*, 149:110220, 2024. 2, 5

[11] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004. 2

[12] Chenyu Liu, Jia Pan, Jinshui Hu, Baocai Yin, Bing Yin, Mingjun Chen, Cong Liu, Jun Du, and Qingfeng Liu. Namer: Non-autoregressive modeling for handwritten mathematical expression recognition. In *European Conference on Computer Vision*, pages 273–291. Springer, 2024. 2, 5

[13] 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 *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1533–1538. IEEE, 2019. 4

[14] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013. 2

[15] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PmLR, 2016. 2

[16] Harold Mouchere, Christian Viard-Gaudin, Richard Zanibbi, and Utpal Garain. Icfhr 2014 competition on recognition of on-line handwritten mathematical expressions (crohme 2014). In *2014 14th International Conference on Frontiers in Handwriting Recognition*, pages 791–796. IEEE, 2014. 4

[17] Harold Mouchère, Christian Viard-Gaudin, Richard Zanibbi, and Utpal Garain. Icfhr2016 crohme: Competition on recognition of online handwritten mathematical expressions. In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 607–612. IEEE, 2016. 4

[18] 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. 2

[19] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015. 2

[20] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 2

[21] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. 2, 4

[22] Thanh-Nghia Truong, Cuong Tuan Nguyen, Khanh Minh Phan, and Masaki Nakagawa. Improvement of end-to-end offline handwritten mathematical expression recognition by weakly supervised learning. In *2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 181–186. IEEE, 2020. 2, 5

[23] Hashim M Twaakyondo and Masayuki Okamoto. Structure analysis and recognition of mathematical expressions. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, pages 430–437. IEEE, 1995. 2

[24] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, and Nando Freitas. Dueling network architectures

for deep reinforcement learning. In *International conference on machine learning*, pages 1995–2003. PMLR, 2016. 2

[25] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256, 1992. 2

[26] Changjie Wu, Jun Du, Yunqing Li, Jianshu Zhang, Chen Yang, Bo Ren, and Yiqing Hu. Tdv2: A novel tree-structured decoder for offline mathematical expression recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2694–2702, 2022. 1, 2, 5

[27] Jin-Wen Wu, Fei Yin, Yan-Ming Zhang, Xu-Yao Zhang, and Cheng-Lin Liu. Image-to-markup generation via paired adversarial learning. In *Machine learning and knowledge discovery in databases: European conference, ECML PKDD 2018, Dublin, Ireland, September 10–14, 2018, Proceedings, Part I 18*, pages 18–34. Springer, 2019. 2, 5

[28] Jin-Wen Wu, Fei Yin, Yan-Ming Zhang, Xu-Yao Zhang, and Cheng-Lin Liu. Handwritten mathematical expression recognition via paired adversarial learning. *International Journal of Computer Vision*, 128:2386–2401, 2020. 2, 5

[29] Wentao Yang, Zhe Li, Dezhi Peng, Lianwen Jin, Mengchao He, and Cong Yao. Read ten lines at one glance: line-aware semi-autoregressive transformer for multi-line handwritten mathematical expression recognition. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 2066–2077, 2023. 2, 4, 5

[30] Zhe Yang, Qi Liu, Kai Zhang, Shiwei Tong, and Enhong Chen. Gap: a grammar and position-aware framework for efficient recognition of multi-line mathematical formulas. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 901–910, 2024. 2

[31] 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*, pages 4553–4562, 2022. 1, 2, 4, 5

[32] Richard Zanibbi, Dorothea Blostein, and James R. Cordy. Recognizing mathematical expressions using tree transformation. *IEEE Transactions on pattern analysis and machine intelligence*, 24(11):1455–1467, 2002. 2

[33] Jianshu Zhang, Jun Du, Shiliang Zhang, Dan Liu, Yulong Hu, Jinshui Hu, Si Wei, and Lirong Dai. Watch, attend and parse: An end-to-end neural network based approach to handwritten mathematical expression recognition. *Pattern Recognition*, 71:196–206, 2017. 2, 5

[34] Jianshu Zhang, Jun Du, and Lirong Dai. Multi-scale attention with dense encoder for handwritten mathematical expression recognition. In *2018 24th international conference on pattern recognition (ICPR)*, pages 2245–2250. IEEE, 2018. 2, 5

[35] Jianshu Zhang, Jun Du, and Lirong Dai. Track, attend, and parse (tap): An end-to-end framework for online handwritten mathematical expression recognition. *IEEE Transactions on Multimedia*, 21(1):221–233, 2018. 2, 5

[36] Jianshu Zhang, Jun Du, Yongxin Yang, Yi-Zhe Song, and Lirong Dai. Srd: a tree structure based decoder for online handwritten mathematical expression recognition. *IEEE transactions on multimedia*, 23:2471–2480, 2020. 2

[37] 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*, pages 11076–11085. PMLR, 2020. 1, 2, 5

[38] Ting Zhang, Harold Mouchere, and Christian Viard-Gaudin. Tree-based blstm for mathematical expression recognition. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, pages 914–919. IEEE, 2017. 2

[39] Xinyu Zhang, Han Ying, Ye Tao, Youlu Xing, and Guihuan Feng. General category network: Handwritten mathematical expression recognition with coarse-grained recognition task. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023. 2, 5

[40] Wenqi Zhao and Liangcai Gao. Comer: Modeling coverage for transformer-based handwritten mathematical expression recognition. In *European conference on computer vision*, pages 392–408. Springer, 2022. 1, 2, 4, 5

[41] Wenqi Zhao, Liangcai Gao, Zuoyu Yan, Shuai Peng, Lin Du, and Ziyin Zhang. Handwritten mathematical expression recognition with bidirectionally trained transformer. In *Document analysis and recognition–ICDAR 2021: 16th international conference, Lausanne, Switzerland, September 5–10, 2021, proceedings, part II 16*, pages 570–584. Springer, 2021. 1, 2, 4, 5

[42] Shuhan Zhong, Sizhe Song, Guanyao Li, and S-H Gary Chan. A tree-based structure-aware transformer decoder for image-to-markup generation. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 5751–5760, 2022. 2

[43] Jianhua Zhu, Liangcai Gao, and Wenqi Zhao. Ical: Implicit character-aided learning for enhanced handwritten mathematical expression recognition. In *International Conference on Document Analysis and Recognition*, pages 21–37. Springer, 2024. 2, 5

[44] Jianhua Zhu, Wenqi Zhao, Yu Li, Xingjian Hu, and Liangcai Gao. Tamer: Tree-aware transformer for handwritten mathematical expression recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10950–10958, 2025. 2, 5