PosFormer: Recognizing Complex Handwritten Mathematical Expression with Position Forest Transformer (Supplementary Material)

Tongkun Guan[1*], Chengyu Lin[2*], Wei Shen[1(✉)], and Xiaokang Yang[1]

[1] MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
{gtk0615,wei.shen}@sjtu.edu.cn
[2] Paris Elite Institute of Technology, Shanghai Jiao Tong University
lacayqwq@sjtu.edu.cn

1 MNE

1.1 Structural Complexity Definition

The structural complexity of a ME is defined as the maximum nested levels of its substructures (*e.g., the nested level of* $x^{2^2} + x^{2^{2^2}} + x^{2^{2^{2^2}}}$ *is 4*). MEs with nested levels above 2 are considered complex.

1.2 Data construct

We construct a **M**ulti-level **N**ested handwritten mathematical **E**xpression test set, used to evaluate a model's ability to recognize complex expression images. MNE includes three subsets (N1, N2, and N3) with nested levels of 1, 2, and 3, respectively. N4 is ignored because it accounts for only 0.2% of all public datasets, while the other levels are 37.4%, 51.4%, 9.7%, and 1.3%, respectively (statistics). The subsets N1, N2 from CROHME, and N3 were collected by us.

Specifically, we first collect these subsets from the CHOHME test sets, according to the number of nested levels of expressions, including 1875, 304, and 10 images. Subsequently, to more convincingly assess the model's ability to identify complex mathematical expressions, we further expand the sample number of subset N3 to 1464 images by collecting complex expression images from public documents [1–4] and real-world handwriting homework.

1.3 Results on MNE Datasets

As shown in Table 1, we conduct the comparison experiments with the previous SOTA methods on these subsets. The results, derived from loading the optimal checkpoint for single-line dataset performance evaluation, show our method achieving performance gains of 0.86%, 1.65%, and 10.04% on the three subsets, respectively. Moreover, as the complexity of the subsets increases, the performance improvements afforded by PosFormer further escalate, which underscores the significance of enhancing position-aware symbol feature extraction.

Table 1: Performance comparison with previous SOTA methods on the complex MNE test set. ExpRate, $\leq 1, \leq 2, \leq 3$ are shown in percentage (%).

Dataset	Size	Model	ExpRate ↑	≤ 1 ↑	≤ 2 ↑	≤ 3 ↑
N1	1875	COMER	59.73	77.55	84.11	88.91
		PosFormer (ours)	**60.59**$_{(+0.86)}$	**77.97**	**84.32**	**88.75**
N2	304	COMER	37.17	53.95	65.13	72.37
		PosFormer (ours)	**38.82**$_{(+1.65)}$	**56.91**	**66.12**	**73.36**
N3	1464	SAN	8.61	13.18	16.46	20.82
		CAN	7.72	10.52	12.43	15.44
		COMER	24.04	32.31	36.34	39.89
		PosFormer (ours)	**34.08**$_{(+10.04)}$	**36.82**	**40.30**	**43.10**

Table 2: Comparison results of speed and parameter amount.

Model	CoMER	PosFormer	DWAP	DWAP+PF	DWAP+CAN
#Params	6.4M	6.4M	4.7M	4.7M	17.0M
FPS (↑)	6.30	6.28	21.73	21.72	18.47

1.4 Visualization.

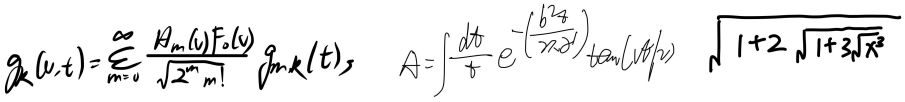

Fig. 1: Some visualization examples.

2 Model Cost

We add speed and parameter comparisons in Table 2. The PF (69K, removed during inference) and IAC (0K) are introduced to **baseline**. We add the comparisons of parameters and latency during inference on the same platform.

3 Difference to Tree-based Methods

Different purposes. Tree-based methods are designed for HMER directly. Position Forest (PF) assists sequence-based methods by explicitly modelling positional relationships between symbols, which is removed during inference and incurs no additional latency or computational cost.

Table 3: Comparison with tree-based methods on CROHME-series datasets. "PF" denotes the Position Forest. ExpRate, $\leq 1, \leq 2$ are shown in percentage (%).

Model	CROHME 2014			CROHME 2016			CROHME 2019		
	ExpRate ↑	$\leq$ **1** ↑	$\leq$ **2** ↑	**ExpRate** ↑	$\leq$ **1** ↑	$\leq$ **2** ↑	**ExpRate** ↑	$\leq$ **1** ↑	$\leq$ **2** ↑
DWAP	50.10	-	-	47.50	-	-	-	-	-
DWAP+SAN	50.41	68.15	76.06	51.87	68.70	75.68	51.13	71.23	79.23
DWAP+PF (ours)	**57.10**$_{(+6.69)}$	**73.02**	**80.53**	**56.23**$_{(+4.36)}$	**72.24**	**81.17**	**57.30**$_{(+6.17)}$	**75.56**	**82.24**

Different mechanisms. Tree-based methods model a mathematical expression as a syntax tree, achieving expression recognition by predicting the entire tree and assembling these entity symbols (nodes) and syntactic relationships (edges). PF models the mathematical expression as a position forest structure, reflecting the actual spatial positions of symbols within the LaTeX expression's substructures. The simple nested levels and relative positions are parsed to positively promote position-aware symbol-level feature extraction. Accordingly, when separately integrating them into sequence-based methods to enhance structural relationship perception, PF can simultaneously predict the category and position for each symbol of the sequence in the same decoder, while the target context of the extra tree-based decoder conflicts with the sequence. We also conduct the ablation experiment in Table 3, comparing with the latest open-source tree-based method SAN, PosFormer brings significant gains.

Explanation Example. Tree-based methods encode an expression as a syntax tree based on complete triplet relationships (parent node, child node, parent-child relations). The absence of any of these relationships will cause encoding failure; *e.g.,* $^{14}CO_2$, as there is no parent node for "1" (`^{14}CO_{2}`). Differently, without strict syntax dependency, the **key** ("^") triggers our PF to view "1" as the upper part in the ME image, thereby encoding the relative position and nested level (we need) is "upper (L)" and 1 to assist training, not depending on other symbols. Moreover, compared to the limitations associated with tree-based methods, PF employs the sequence-based decoding process, where each ME case can be parsed into a sequence.

In summary, usage differences between tree-based methods (A) and our method (B) lie in:

1) Simplify encoding.

(A) ME $\xrightarrow{syntax\ rules}$ a syntax tree $\xrightarrow{target}$ complete triple tuple (parent, child, seven types of parent-child relationships).

(B) ME $\xrightarrow{split}$ substructures $\xrightarrow{M/L/R}$ independent spatial position trees $\xrightarrow{form}$ forest $\xrightarrow{target}$ nested levels, relative positions.

2) Assist decoding. T/I denotes a training/inference stage.

(A) Image $\xrightarrow{stage1}$ triple tuple $\xrightarrow{stage2}$ LaTeX sequence (T/I).

(B) Image $\xrightarrow{one\ stage}$ LaTeX sequence (T/I) + nested levels(**T**) + relative position(**T**), as PF explicitly promotes position-aware symbol-level representation learning at the feature level during training.

References

1. Mouchere, H., Viard-Gaudin, C., Zanibbi, R., Garain, U.: Icfhr 2014 competition on recognition of on-line handwritten mathematical expressions (crohme 2014). In: ICFHR. pp. 791–796 (2014)
2. Mouchère, H., Viard-Gaudin, C., Zanibbi, R., Garain, U.: Icfhr2016 crohme: Competition on recognition of online handwritten mathematical expressions. In: ICFHR. pp. 607–612 (2016)
3. Wang, B., Gu, Z., Xu, C., Zhang, B., Shi, B., He, C.: Unimernet: A universal network for real-world mathematical expression recognition. arXiv preprint arXiv:2404.15254 (2024)
4. Yuan, Y., Liu, X., Dikubab, W., Liu, H., Ji, Z., Wu, Z., Bai, X.: Syntax-aware network for handwritten mathematical expression recognition. In: CVPR. pp. 4553–4562 (2022)