A Hybrid Vision Transformer Approach for Mathematical Expression Recognition

Anh Duy Le*, Van Linh Pham*, Vinh Loi Ly*, Nam Quan Nguyen*, Huu Thang Nguyen*, Tuan Anh Tran†‡(✉)
* Viettel Cyberspace Center, Viettel Group, Vietnam.
Lot D26 Cau Giay New Urban Area, Yen Hoa Ward, Cau Giay District, Hanoi, Vietnam.
† Faculty of Computer Science & Engineering, Ho Chi Minh City-University of Technology (HCMUT),
268 Ly Thuong Kiet Street, District 10, Ho Chi Minh City, Vietnam.
‡ Vietnam National University Ho Chi Minh City, Linh Trung Ward, Thu Duc District, Ho Chi Minh City, Vietnam.
{leanhduy497, phamvanlinh143, vinhloiit1327, ngnamquan}@gmail.com, nhthang99@outlook.com, trtanh@hcmut.edu.vn

2022 International Conference on Digital Image Computing: Techniques and Applications (DICTA) | 978-1-6654-5642-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/DICTA56598.2022.10034626

***Abstract*—One of the crucial challenges taken in document analysis is mathematical expression recognition. Unlike text recognition which only focuses on one-dimensional structure images, mathematical expression recognition is a much more complicated problem because of its two-dimensional structure and different symbol size. In this paper, we propose using a Hybrid Vision Transformer (HVT) with 2D positional encoding as the encoder to extract the complex relationship between symbols from the image. A coverage attention decoder is used to better track attention's history to handle the under-parsing and over-parsing problems. We also showed the benefit of using the [CLS] token of ViT as the initial embedding of the decoder. Experiments performed on the IM2LATEX-100K dataset have shown the effectiveness of our method by achieving a BLEU score of 89.94 and outperforming current state-of-the-art methods.**

***Index Terms*—Mathematical Expression Recognition, Vision Transformer, Encoder-Decoder, OCR**

I. Introduction

Mathematical expression recognition is one of the important processes in scientific documents analysis [1]. Despite the importance of this task, solving mathematical expression recognition is still very challenging. One of the reasons for the difficulty of math recognition compared to normal text recognition is that math formula usually has 2-D spatial structure relationship [2] instead of 1-D ones from normal text data. The spatial structure relationship of math formula is presented by many math symbols such as superscript, subscript, fraction symbol, etc. The traditional approach usually solves this problem in two stages. First, the character segmentation stage is used to segment each character in math formula and then classify it based on the given vocabulary. Second, the structural analysis stage is used to identify the spatial relationships between all characters of the math formula.

Due to the success of sequence to sequence (Seq2seq) architecture [3] from machine translation problems, many recent works have applied this architecture to many other fields, including speech recognition [4], text recognition [5], image captioning [6]. Seq2seq architecture includes two main parts: encoder and decoder. Mathematical expression recognition can also be considered a sequence translation problem, where the input, in this case, is the image of math expression and the output is a 1-D sequence of LaTeX. Therefore, we can use the Seq2seq approach to solve the mathematical expression recognition problem. Indeed, recent works have proposed many variants of Seq2seq architecture [2], [7]–[10] and achieved many promising results. Despite that, the design of these architectures still has many limitations. For example, Deng et al. [7] introduced a multi-row encoder to capture the non-left-to-right relationships of math symbols better, or Zhang et al. [8] with a multi-scale encoder using DenseNet [11] to handle the different size of symbols. Recently, Zelun Wang and Jyh Charn Liu [12] had designed a convolutional neural network (CNN) backbone with an additional 2D positional encoding and performed sequence-level learning based on reinforcement learning. These models entirely depend on the feature extracted by a CNN. They so lack global information, which is necessary for modeling spatial relationships between different math symbols since math expressions can contain related symbols which are far apart, limiting them in recognizing long expressions. Inspired by the success of Vision Transformer (ViT) [13] architecture, we propose a novel Hybrid Vision Transformer (HVT) approach to acting as an encoder of the Seq2seq model to alleviate the lacking of global information problem. An HVT consists of a CNN backbone, a 2D positional encoding (2DPE), and a stacking of multiple ViT blocks. Image's features are first extracted by a CNN backbone to reduce input size and get high-level information then are encoded with global information using ViT to return the annotation vectors [14]. 2D positional encoding helps the feature maps reserve more spatial information for both vertical and horizontal directions. For the decoder, we follow

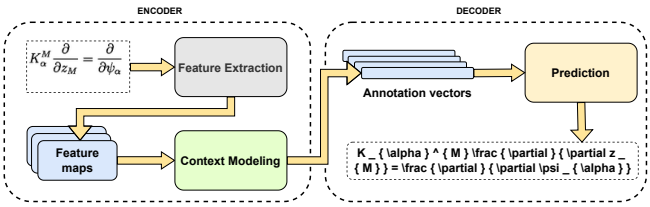


Fig. 1. A general overview of our proposed framework

the coverage attention idea from [14] by using an additional coverage vector to align the attention weights. Furthermore, we also leverage the [CLS] token embedding of ViT as the initial hidden state for our decoder. In general, our architecture includes three main stages as shown in Fig. 1: Feature extraction and context modeling using HVT in the encoder and prediction in the decoder. Experiments on benchmark dataset IM2LATEX-100K have shown a competitive result and achieved a new state-of-the-art (SOTA) result with a BLEU score of 89.94, an image exact match rate of 86.48. Our main contributions can be summarized as follows:

A novel Hybrid Vision Transformer approach for the encoder of the Seq2seq model.

Re-design the Seq2seq framework in both the encoder and the decoder to better suit the math recognition problems and achieve the SOTA result on the IM2LATEX-100K dataset.

Extensive ablation experiments and analysis.

Our remaining sections are organized as follows: Review the related work, present our proposed method, perform experiments and analyze results, and give conclusions and future work.

II. Related work

Mathematical expression recognition has been an interesting research topic for a long time. Before the rise of the deep learning era, researchers usually proposed new methods based on two main approaches: rule-based and grammar-based methods. These methods require knowledge about mathematical grammar and tedious work to design suitable rules. In the theory point of view, mathematical expression recognition problem can be considered a image-based sequence prediction problem which can be solved by a Seq2seq model. With the rise of deep learning, many Seq2seq models [3] which learn directly from data are being proposed and achieved better performance than previous non-deep methods. Deng et al. [7] is considered the first paper to use the Seq2seq model for this problem. In their work, they proposed to use a multi-row encoder to learn the spatial structure of math formulas better. As an improvement from Deng et al., Zhang et al. [14] integrate a coverage vector into the attention module to deal with over parsing in mathematical expression recognition. The coverage vector gives attention module information about the history of alignment in the past to push attention weight to appropriate local regions of feature maps. Zhang et al. [8] have introduced a novel multi-scale encoder to better capture different sizes of math symbols, usually in handwriting math expressions. Using a multi-scale encoder, the model can learn the representation of large and small symbols. Bender et al. [15] focused on extracting fine-grained features from math images. In order to handle the neglect of key features when perform attention causing the decoder to give wrong prediction, Li et al. [16] used a drop attention module to randomly suppress features in training phase, thus, make the model more robust. Yan et al. [2] proposed a decoder that used a CNN instead of the recurrent network to speed up the training and predicting process. Pang et al. [10] use a global-context network to aggregate different global features using a global context module integrated into a CNN backbone. Their backbone is not optimized for text-based recognition compared to ours. It has a more well-designed CNN backbone for math recognition and is entirely based on ViT to capture global dependencies and position information.

III. Methodology

A. Problem definition

Mathematical expression recognition can be considered as a sequence prediction problem where the input is a grayscale image $\mathbf{X} \in \mathbb{R}^{H \times W}$ and the LaTeX ground truth sequence $\mathbf{Y} = \{y_1, y_2, \cdots, y_\tau\}$ with vocabulary of size K and the length of the sequence is τ. The goal of our model is to convert the input image into the corresponding LaTeX sequence by finding a mapping function f such that $f(\mathbf{X}) = \mathbf{Y}$. In practice, we can just find a function f' which is an approximation of function f. In this paper, we achieve this purpose by training our model using a dataset of pairs of the image-LaTeX sequences in a supervised manner. Fig. 2a demonstrates our proposed architecture including HVT for the encoder and coverage attention for the decoder.

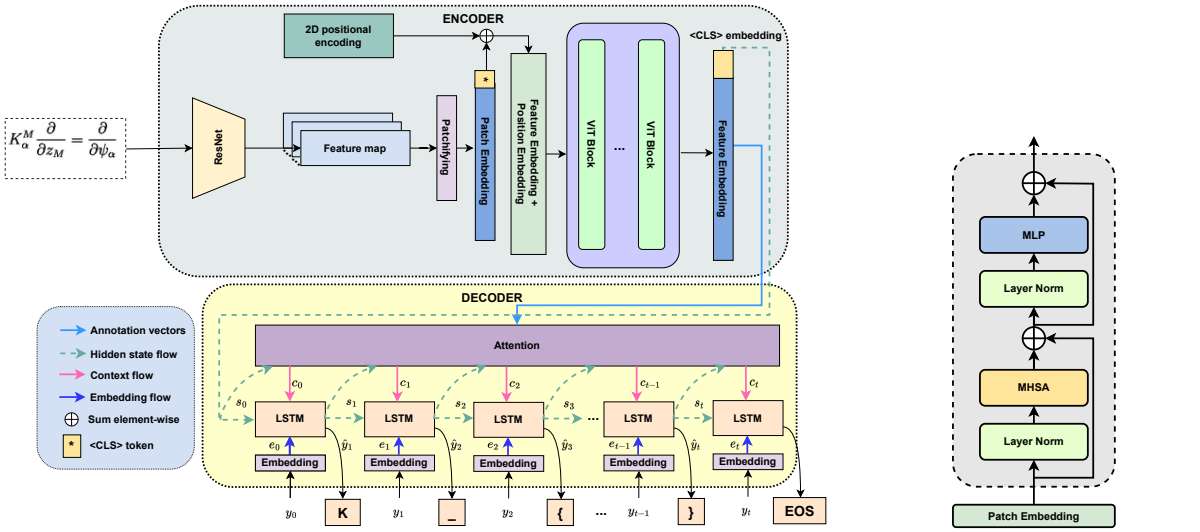
B. Hybrid Vision Transformer as encoder

Our HVT consists of two modules: First, a CNN is considered as a backbone to extract high-level feature from input image. Second, a context modeling module consists of many ViT block stacks together to further enhance feature embedding by modeling global information and capture long-range dependencies between different feature of the feature maps.

The ViT can perform feature representation of image data for image classification task as shown in [13] by their capability of learning internal relationship between pixels in the images using self-attention mechanism without the need of stacking multiple CNN layers. Therefore, ViT can be served as a perfect encoder in the encoder-decoder framework. While recent works are still based on a recurrent neural network to model context information from the feature maps, one of the most commonly used is bidirectional LSTM (BiLSTM) which can combine the context in both directions of the feature maps. However, BiLSTM will become a bottleneck of the whole architecture due to the sequential design.

Due to the missing of inductive bias in localization compared to CNN as claimed in [13], ViT needs a lot of training data to attend on small distances like CNN [17]. To make our model converge easily, we add a supporting CNN backbone before the ViT blocks to encode the image's local regions into high-level features.

1) Backbone: We design our ResNet model based on [18]. Our ResNet-based backbone consists of 32 layers with 4 ResNet blocks. In order to handle text-based images appropriately, stride values at third pooling layer and sixth convolutional layer are changed to (1,2) instead of (2,2) so



(a) The pipeline of proposed architecture

(b) An illustration of ViT block

Fig. 2. The proposed architecture pipeline and illustration of ViT block.

that feature maps can have a larger width making them easier to cover a correct receptive field of a symbol.

Specifically, give an image $\mathbf{X} \in \mathbb{R}^{H \times W}$, the corresponding output feature maps is $\mathcal{F} \in \mathbb{R}^{H_f \times W_f \times C}$, where $H_f = \frac{H}{32} + 1$, $W_f = \frac{W}{4} + 1$, C is number of output channels.

2) Vision Transformer: We follow the standard design of ViT from [13] as shown in Fig. 2b, such that each ViT block contains a self-attention layer to calculate attention probabilities between *query* vector $\mathbf{q}$ and *key* matrix $\mathbf{K}$ and a feed-forward-network (FFN) which consists of two multilayer perceptron layers (MLP). In order to pay attention to different subspace of different symbols' positions, we further apply multi-head self-attention instead of single-head. Similar to the original Transformer [19], ViT uses LayerNorm (LN) [20] to stabilize the learning process. Specifically, given $\mathcal{F} \in \mathbb{R}^{H_f \times W_f \times C}$ as the output feature map from the backbone, in order to convert $\mathcal{F}$ to the correspond representation of input of ViT which is a sequence of 1D token embeddings, we further apply a CNN layer with kernel size of $p \times p$ and strides of $p \times p$, where p is defined as the patch size of a patch image, to create a new feature map $\overline{\mathcal{F}} \in \mathbb{R}^{\frac{H_f}{p} \times \frac{W_f}{p} \times D}$, where D is embedding dimensions. The result is finally flattened into a sequence of tokens $\mathbf{E} \in \mathbb{R}^{N \times D}$ known as patch embeddings with $N = \frac{H_f \times W_f}{p^2}$ is the number of patches.
ViT also provides an additional token $\mathbf{x}_{class}$ as a learnable embedding which is called [CLS] token inspired by the idea used in BERT [21], [CLS] token is concatenated with other spatial tokens in the current patch embeddings $\mathbf{E}$, in the training process, information is forced to flow from all other tokens to [CLS] token through a self-attention mechanism, thus [CLS] token embedding can be considered as a global representation of image features and can be used as an initial hidden state for model's decoder instead of using whole encoder's output feature maps. Moreover, due to the permutation-invariant of self-attention that treats all tokens in sequence as bag-of-word, positional embeddings $\mathbf{E_{pos}}$ with the same dimension D as patch embeddings are incorporated with patch embeddings to provide position information. The final vector $\mathbf{H_0}$ calculated by Eqs. 1 is considered as the input vector to ViT.

$$\mathbf{H}_0 = [\mathbf{x}_{\text{class}}, \mathbf{E}] + \mathbf{E}_{\text{pos}} \tag{1}$$

Where $\mathbf{E} \in \mathbb{R}^{N \times D}, \mathbf{E}_{pos} \in \mathbb{R}^{(N+1) \times D}$. Given L ViT blocks of our HVT, Eq. 2 performs two general steps in the total process of a single ViT block. The MHSA layer first processes the input sequence to mix the information in all tokens in a global context manner. Therefore, one token can accumulate information about other tokens' spatial or semantic features throughout the training process. In mathematical expression recognition, MHSA helps model learn to extract spatial structure for structure analysis and semantic information for symbol recognition. The second step of this process is to go through an FFN. An FFN combines two MLP layers, one layer transforms hidden embeddings from D to 4D dimension and one layer converts 4D back to D dimension. The FFN helps integrate information independently in each token of the sequence.

Concretely, given input sequence $\mathbf{H}_0 \in \mathbb{R}^{(N+1) \times D}$ we first project it into query $\mathbf{Q}^h$, key $\mathbf{K}^h$, value $\mathbf{V}^h$ matrix for each head $h \in [1, N_{head}]$ of MHSA layer using learnable matrix $W_Q^h \in \mathbb{R}^{d^q \times D}$, $W_K^h \in \mathbb{R}^{d^k \times D}$, $W_V^h \in \mathbb{R}^{d^v \times D}$, in our case $d^q = d^k = d^v = \frac{D}{N_{head}}$. An MHSA layer, as shown in Eq. 3 performs self-attention on multiple heads and concatenates the result of all heads together, then projects back to the D dimension using $W_O \in \mathbb{R}^{N_{head} \cdot d^v \times D}$. The final output in Eq. 4 consists of N + 1 elements, including N spatial embedding vectors of the image $\{h_L^1, h_L^2, \cdots, h_L^N\}$ called annotation vectors $\mathbf{A}$ and [CLS] token embedding (i.e. h_L^0).

$$\begin{cases} \mathbf{H}'_\ell = \mathrm{MHSA}\left(\mathrm{LN}\left(\mathbf{H}_{\ell-1}\right)\right) + \mathbf{H}_{\ell-1} \\ \mathbf{H}_\ell = \mathrm{FFN}\left(\mathrm{LN}\left(\mathbf{H}'_\ell\right)\right) + \mathbf{H}'_\ell \end{cases} \quad \ell \in [1, L] \tag{2}$$

$$\mathrm{MHSA}_\ell(\mathrm{Q}, \mathrm{K}, \mathrm{V}) = \left[SA^1_\ell, SA^2_\ell, \cdots, SA^H_\ell\right] \times W_O \tag{3}$$

$$\left[h^0_L, h^1_L, h^2_L, \cdots, h^N_L\right] = \mathbf{H}_L \tag{4}$$

3) 2D Positional Encoding: Unlike the natural image, a math image has a strong semantic correlation between different components in the formula, such position information about nested or hierarchical components needs to be reserved carefully. Under this assumption, we propose 2DPE which using 2D sinusoidal positional encoding similar to [22]. 2DPE performs 1D positional encoding from [19] on two dimensions of the input features and then concatenates them to gain the final output.

Concretely, (H, W) is height and width of the feature maps and D is embedding dimension, the position embedding of the feature maps is calculated by our 2DPE using Eq. 5:

$$\mathrm{2DPE}(H, W, D) = [\mathrm{PE}(h, i), \mathrm{PE}(w, j)] \tag{5}$$

Where $i \in [0, D/2), j \in [D/2, D)$, [.] denotes concatenate operation. The embedding for each dimension can be obtained as follow:

$$\begin{cases} \mathrm{PE}\left(pos, t\right) = \sin\left(\mathrm{pos}\,/10000^{t/\frac{D}{4}}\right) \\ \mathrm{PE}\left(pos, t + \frac{D}{4}\right) = \cos\left(\mathrm{pos}\,/10000^{t/\frac{D}{4}}\right) \end{cases} \tag{6}$$

Where $t \in [0, D/4)$, PE denotes 1D sinusoidal positional encoding, and pos denotes any position in the horizontal (w) or vertical direction (h).

C. Coverage attention for Decoder

To deal with variable input length of math image and variable output length of LaTeX sequence, we choose RNN-based with attention mechanism [23] as our decoder. At every time step t, our decoder calculate a fixed-size intermediate vector also known as context vector $\mathbf{c}_t \in \mathbb{R}^D$ based on a weighted sum between the annotation vectors $\mathbf{A}$ and the attention weight $\boldsymbol{\alpha}_t \in \mathbb{R}^N$. By using $\mathbf{c}_t$, our decoder can generate a suitable LaTeX symbol and also be independent of variable input-output length. We consider using unidirectional LSTM instead of traditional recurrent neural networks to overcome the vanishing gradient problem.

At training phase, each groundtruth (GT) token $\mathbf{y}$ is first transformed using an embedding layer as shown in Fig. 2 into a vector $\mathbf{e}$ so that tokens which have high correlation (e.g. '[', ']') can have similar embedding.

At time step 0, we choose vector embedding of [CLS] token (i.e. h^0_L) as the initial hidden state. Different from [24] approach, we further apply an MLP layer to h^0_L to integrate more information. Coverage attention similar to [14] is applied to our decoder to make it more robust with under-parsing and over-parsing problems. Coverage vector is created by summing all previous attention weights $\boldsymbol{\alpha}$ and performing a convolution operation (Conv) to aggregate information of the history alignments as shown in Eq. 7 and therefore it can guide the decoder on future prediction to put attention on appropriate regions.

$$\begin{cases} \boldsymbol{\beta}_t = \sum_{l=1}^{t-1} \boldsymbol{\alpha}_l \\ \mathbf{f}_t = \mathrm{Conv}(\boldsymbol{\beta}_t) \end{cases} \tag{7}$$

Where $\mathbf{f}_t$ indicates coverage vector at step t, β_0 is set to vector 0. In summary, the output $\hat{\mathbf{y}}_t \in \mathbb{R}^K$ at step t is calculated by Eq. 8.

$$\begin{cases} \mathbf{c_t} = \mathrm{Attention}(\mathbf{s}_{t-1}, \mathbf{A}, \mathbf{f}_t)) \\ \mathbf{s}_t = \mathrm{RNN}(\mathbf{s}_{t-1}, [\mathbf{c}_t, \mathbf{e}_t]) \\ \hat{\mathbf{y}}_\mathbf{t} = \mathrm{Softmax}(\mathbf{s}_t) \end{cases} \tag{8}$$

Where $\mathrm{Attention}$ is attention operation as described in [14], RNN is a recurrent layer specifically a LSTM layer, $\mathrm{Softmax}$ is a softmax layer to output probability distribution, [.] denotes concatenate operation, $\mathbf{s}_j$ is decoder hidden state of step $j \in [0, \tau]$, $\mathbf{e}_t$ is embedding vector of $\mathbf{y}_t$ GT token. At time step 0, $\mathbf{s}_0 = \mathrm{MLP}(\mathbf{h}^0_L)$.

IV. Experiments

This section presents our experiment on the benchmark dataset and compares it with other SOTA models. We also perform an extensive ablation study to analyze the effectiveness of each component in our model, and finally, some visualization of the attention weight in both encoder and decoder.

A. Dataset

We choose to use a benchmark dataset IM2LATEX-100k [25] created by crawling from 60000 research papers on arXiv. The dataset contains 103,556 LaTeX representation of mathematical expressions in total. The length of each LaTeX sequence is change from 38 to 997. Each sequence is rendered to PDF format using *pdflatex* compiler and converted to PNG image in grayscale format using *ImageMagick*. The final dataset is split into 3 partitions including train set with 83,883 formulas, validation set with 9319 formulas, and test set with 10,354 formulas.

We follow the same preprocessing strategy as [7] by applying a parsing algorithm on raw LaTeX sources to create tokenized LaTeX labels. We then create a vocabulary V from all unique LaTeX tokens with three addition tokens, including [SOS], [EOS], and [PAD] tokens. Our vocabulary V contains 499 tokens. All images of the same size will be grouped into the same bucket. This way can help to reserve the 2D structure of math images, which is different from normal text images.

B. Implementation details

1) Architecture: For our HVT configuration, we set the number of output channels of ResNet-based backbone to $C = 512$. Different from the original setting of ViT from [13], we reduce the number of heads to $N_{head} = 8$ and we only use $L = 6$ ViT blocks to encode the image's feature, we choose the dimension the patch embedding to $D = 256$, dimension

of FFN layer to $d_{ffn} = 1024$. ince we perform patchify on feature maps, we choose a small patch size $p = 2$. For the decoder, we apply a filter with a kernel size of 5×5 and an output channel of 128 to the coverage vector; both the hidden state of LSTM and the embedding size of the input token are set to $d_{emb} = 256$. We adopt a dropout [26] technique with drop rate 0.1 as a regularization method to reduce over-fitting.

2) Training and inference: In the training phase, at every time step, the decoder will receive the embedding of the ground truth token, also known as teacher forcing. In the inference phase, the output LaTeX sequence is predicted token by token at every time step. Moreover, to prevent the predicted output from sampling from a sub-optimal distribution, we use beam search with beam size set to 5 to get the output token. The entire model is trained from scratch without using any pretrained weight for 300K iterations with a batch size of 32 using AdamW [27] optimizer with an initial learning rate set to 5×10^{-4} and the decay rate of 2×10^{-6}. After every step, the learning rate is adjusted by using a warm-up cosine schedule. A simple data augmentation strategy that includes random scale and rotation is adopted to better optimize ViT. All experiments are implemented using PyTorch and conducted on GPU NVIDIA V100 32GB.

3) Evaluation: We also consider evaluating our model's performance using text- and image-based metrics. For text-based metrics, we use BLEU-4 score [28], text edit distance (TED) which compute the Levenshtein distance between the GT sequence and the prediction at token level, and the sequence accuracy (Acc) which return 1 if the prediction and GT are exactly the same else 0. For image-based metrics, we evaluate on the rendered images of predicted LaTeX sequences and GT images using the same metrics used in [7] including image edit distance (IED), and exact match accuracy without spaces (EMA w/o space).

C. Compare between other SOTA methods

We demonstrate the effectiveness of our proposed model by comparing it with previous methods on IM2LATEX-100K test set. Our method achieves a better result than [10] which proves the effectiveness of using ViT compared to a global context module from [10]. Especially, for image-base metric as EMA, our method obtain a significant improvement of about 2.4% compared to [9] which proves the potential of our method in capturing the image's structure. Besides, Acc result which is very low compare to other metrics has shown the ambiguities in LaTeX grammar where many LaTeX sequences can represent the same visual structure.

D. Ablation Study

1) Contribution of main components: To better understand the effectiveness of each component in our proposed method, we extensively compare the performance of the baseline model with other model's versions when we alternately replace each component of the baseline with our proposed ones.
For the baseline, we choose VGG [29] architecture as the backbone and do not use any context modeling module, attention mechanism (Attn) without coverage vector similar to [23] is used as the prediction head. For feature extraction, we compare VGG backbone with our ResNet-based. For context modeling, we choose BiLSTM from [23], we also experiment on a ViT version which receives 1D feature maps called ViT-1D by collapsing the height of input feature maps into one together with our's ViT-2D to perform the comparison. For prediction, we consider using a transformer decoder from [19] and our coverage attention (Coverage-Attn) against the baseline. Detail of the all experiment setting and the results of ablation experiments are shown in Table II and III.
According to the results, using our ResNet-based instead of VGG as the backbone has improved the baseline by a large margin on all evaluation metrics, especially the accuracy has increased by 15%. This suggests that a ResNet model for text recognition is a good choice for our backbone. In the context modeling component, we can observe that adding the ViT-2D module gives the best performance. It shows the ability to model a better long-range dependencies compared to BiLSTM and reserve a 2D spatial structure compared to ViT-1D. Using coverage attention for the baseline instead of [23] also gives a better result than using the transformer decoder, which indicates the importance of the coverage vector in keeping the alignment history.

TABLE I
COMPARISON BETWEEN DIFFERENT METHODS ON IM2LATEX-100K.

Method	Acc	BLEU-4	EDA	EMA (w/o space)
Global Context [10]	-	89.72	90.07	82.13
Double Attention [9]	-	89.4	90.9	84.1
MI2LATEX w/o reinforce [12]*	-	89.08	91.09	82.13
Ours	**48.39**	**89.94**	**92.23**	**86.48**

* We only consider the MI2LATEX version without second training phase.

TABLE II
DETAIL OF ABLATION EXPERIMENT SETTING, WHERE 'NONE' INDICATE EMPTY MODULE.

Experiment	Encoder		Pred.
	Feat.	Context.	
Baseline	VGG	None	Attn
V1	ResNet	None	Attn
V2	VGG	ViT-1D	Attn
V3	VGG	BiLSTM	Attn
V4	VGG	ViT-2D	Attn
V5	VGG	None	Transformer Decoder
V6	VGG	None	Coverage-Attn
V7	ResNet	ViT-2D	Coverage-Attn

2) Contribution of 2d positional encoding: To study the impact of 2D positional encoding, we compare the result when using 1D positional encoding similar to [13], [19] and our 2DPE before entering the ViT. Table IV shows that using our 2DPE has a better result on all metrics. Significantly, the EMA

TABLE III
ABLATION EXPERIMENTS ON DIFFERENT KEY COMPONENTS OF OUR APPROACH EVALUATE ON IM2LATEX-100K VALIDATION SET. PARTICULARLY, FEAT. DENOTE FEATURE EXTRACTION, CONTEXT. DENOTE CONTEXT MODELING, AND PRED. DENOTE PREDICTION.

Component	Experiment	Acc %	BLEU-4 %	TED %	params $\times 10^6$
Feat.	Baseline	22.15	79.12	77.32	7.05
	V1	**37.55**	**85.90**	**89.90**	**45.76**
Context.	Baseline	22.15	79.12	77.32	7.05
	V2	34.97	87.73	92.23	12.13
	V3	43.69	90.61	94.52	9.75
	V4	**44.42**	**91.48**	**94.88**	**12.19**
Pred.	Baseline	22.15	79.12	77.32	7.05
	V5	42.06	90.12	93.04	**15.52**
	V6	**42.58**	**90.23**	**93.12**	7.09
Overall	**V7**	**49.30**	**92.33**	**95.60**	**50.95**

has improved by approximately 4%. Experiment's result is evaluated on IM2LATEX-100K test set.

TABLE IV
COMPARISON BETWEEN TWO POSITIONAL ENCODING METHOD WHEN APPLY TO OUR PROPOSED METHOD.

Positional Encoding	Acc	BLEU-4	IED	EMA (w/o space)
1D	45.78	88.84	89.31	82.93
2D	**48.39**	**89.94**	**92.23**	**86.48**

3) Contribution of [CLS] token embedding vector: To study the impact of using [CLS] token embedding as an initial hidden state for the decoder, we compare the result when not using initial embedding and when using it. Table V shows that 2D positional encoding has obtained a better result than the 1D positional encoding approach. Experiment's result is evaluated on IM2LATEX-100K test set.

TABLE V
COMPARISON ON TWO INITIAL HIDDEN STATE SETTINGS WHEN APPLY TO OUR PROPOSED METHOD.

Use init	Acc	BLEU-4	IED	EMA (w/o space)
✓	**48.39**	**89.94**	**92.23**	**86.48**
✗	43.87	81.73	89.36	81.02

E. Discussion

1) The effect of LaTeX sequence length: Given the ground truth of LaTeX sequences, we manually group them into different groups based on their length to investigate the effect of LaTeX sequence length on the performance of our method. To prove the consistency of our method to the sequence length, we compare the average EMA (w/o space) between our method and the baseline model on different group lengths. Fig. 3 shows that our method is very robust to the length of LaTeX sequences while the baseline's performance decreases significantly. Our method still has the EMA of more than 71% when sequence length is more significant than 100 and has 26% for a sequence with more than 150 tokens.

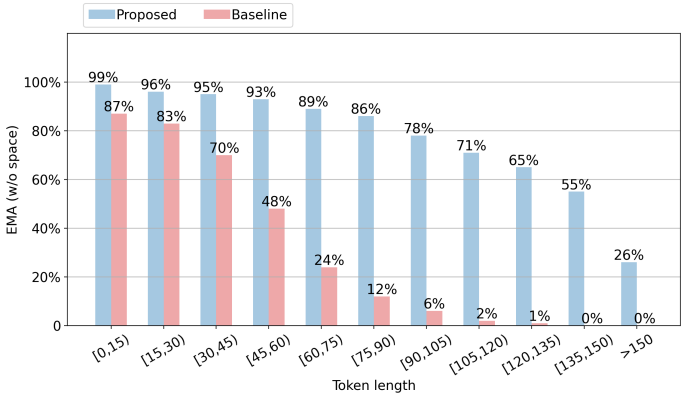


Fig. 3. Comparison between our proposed method and the baseline model at different sequence length.

2) Encoder visualization: The input of ViT is the sequence of embedding vector of spatial locations in the input image plus the embedding of [CLS] token. Fig. 4 shows the self-attention map of the [CLS] token embedding when attending to all other spatial embedding vectors. This visualization has confirmed the usefulness of using our HVT in modeling global information between different math symbols in the images.

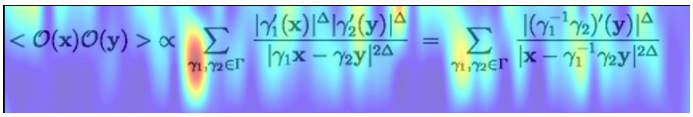


Fig. 4. Examples of self-attention map of [CLS] token embedding.

3) Decoder visualization: We visualize the step-by-step decoding process using coverage attention on an example math expressions of IM2LATEX-100K testing in Fig. 5. At each step, the attention map shows that the model correctly aligns some local region on the image to the corresponding math symbol.

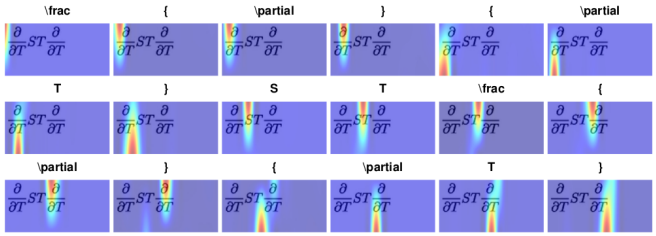
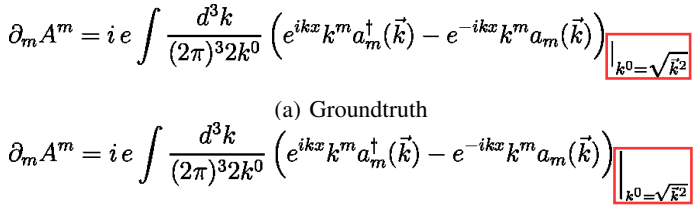


Fig. 5. Visualization of a step-by-step decoding process using our method on example math expression image.

4) Limitation: Fig. 6 has shown that despite the strong ability to capture the global dependencies and correlation between symbols in math image, and the capable symbol recognition mechanism through coverage attention help the model to implicitly learn about the grammar rules, it still suffers from the lack of specific knowledge about grammar.



(a) Groundtruth

(b) Prediction

Fig. 6. An example of our model's prediction about the correctness in spatial structure and symbol correlation but misunderstanding in the syntactic relationships.

V. Conclusion and Future Work

In this paper, we have proposed a novel Hybrid Vision Transformer approach combined with the embedding vector of [CLS] token as the initial hidden state of the decoder allowing the model to extract more sophisticated relationships. Our architecture includes three main stages, which are feature extraction, context modeling and prediction. Our approach has proved the effectiveness when compared to other approaches. Our model has achieved a SOTA performance on the well-know public dataset IM2LATEX-100K. In the future, our research will focus on appending synthetic LaTeX information into the Seq2seq model to better handle more complicated math expression structure. Besides, we will build a complete system to be able to provide products to users.

Acknowledgment

We acknowledge Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for supporting this study.

References

[1] T. A. Tran, K. Oh, I.-S. Na, G.-S. Lee, H.-J. Yang, and S.-H. Kim, "A robust system for document layout analysis using multilevel homogeneity structure," *Expert Systems With Applications*, vol. 85, no. 1, pp. 99–113, 2017.

[2] Z. Yan, X. Zhang, L. Gao, K. Yuan, and Z. Tang, "Convmath: A convolutional sequence network for mathematical expression recognition," in *25th ICPR*, 2021, pp. 4566–4572.

[3] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in neural information processing systems*, vol. 27, 2014.

[4] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell," *arXiv preprint arXiv:1508.01211*, 2015.

[5] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE TPAMI*, vol. 39, no. 11, pp. 2298–2304, 2016.

[6] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *32nd ICML*. PMLR, 2015, pp. 2048–2057.

[7] Y. Deng, A. Kanervisto, and A. M. Rush, "What you get is what you see: A visual markup decompiler," *arXiv preprint arXiv:1609.04938*, vol. 10, pp. 32–37, 2016.

[8] J. Zhang, J. Du, and L. Dai, "Multi-scale attention with dense encoder for handwritten mathematical expression recognition," in *24th ICPR*. IEEE, 2018, pp. 2245–2250.

[9] W. Zhang, Z. Bai, and Y. Zhu, "An improved approach based on cnn-rnns for mathematical expression recognition," in *4th International Conference on Multimedia Systems and Signal Processing*, 2019, pp. 57–61.

[10] N. Pang, C. Yang, X. Zhu, J. Li, and X.-C. Yin, "Global context-based network with transformer for image2latex," in *25th ICPR*. IEEE, 2021, pp. 4650–4656.

[11] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *CVPR*, 2017, pp. 4700–4708.

[12] Z. Wang and J.-C. Liu, "Translating math formula images to latex sequences using deep neural networks with sequence-level training," *IJDAR*, vol. 24, no. 1, pp. 63–75, 2021.

[13] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[14] J. Zhang, J. Du, S. Zhang, D. Liu, Y. Hu, J. Hu, S. Wei, and L. Dai, "Watch, attend and parse: An end-to-end neural network based approach to handwritten mathematical expression recognition," *Pattern Recognition*, vol. 71, pp. 196–206, 2017.

[15] S. Bender, M. Haurilet, A. Roitberg, and R. Stiefelhagen, "Learning fine-grained image representations for mathematical expression recognition," in *ICDARW*, vol. 1. IEEE, 2019, pp. 56–61.

[16] Z. Li, L. Jin, S. Lai, and Y. Zhu, "Improving attention-based handwritten mathematical expression recognition with scale augmentation and drop attention," in *2020 17th ICFHR*. IEEE, 2020, pp. 175–180.

[17] M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, and A. Dosovitskiy, "Do vision transformers see like convolutional neural networks?" *Advances in Neural Information Processing Systems*, vol. 34, pp. 12 116–12 128, 2021.

[18] Z. Cheng, F. Bai, Y. Xu, G. Zheng, S. Pu, and S. Zhou, "Focusing attention: Towards accurate text recognition in natural images," in *ICCV*, 2017, pp. 5076–5084.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[20] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[21] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT*, 2018.

[22] X. Chen, S. Xie, and K. He, "An empirical study of training self-supervised vision transformers," in *ICCV*, 2021, pp. 9640–9649.

[23] B. Shi, M. Yang, X. Wang, P. Lyu, C. Yao, and X. Bai, "Aster: An attentional scene text recognizer with flexible rectification," *TPAMI*, vol. 41, no. 9, pp. 2035–2048, 2018.

[24] Y. Tao, Z. Jia, R. Ma, and S. Xu, "Trig: Transformer-based text recognizer with initial embedding guidance," *Electronics*, vol. 10, no. 22, p. 2780, 2021.

[25] Y. Deng, A. Kanervisto, J. Ling, and A. M. Rush, "Image-to-markup generation with coarse-to-fine attention," in *ICML*. PMLR, 2017, pp. 980–989.

[26] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[27] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *ICLR*, 2019.

[28] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.

[29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.