Handwritten Mathematical Expressions Recognition via Mutual learning

Anonymous submission
PaPer ID:

Abstract

Handwritten mathematical expression recognition (HMER) aims to automatically generate the LaTex form given an input expression image. Currently, The popular models based on the attention mechanism face the problem of over-parsing and under-parsing, because of the imperfect attention on complex two-dimensional structure of HM. In this paper, we propose Cooperative Attentive Translation Network (CATNet) to improve the accuracy of feature attention by the mutual learning frame of using multi-branches decoders. We continuously enhance the attention of both two branches, which involves synchronous executions of two one-to-one knowledge transfers at each decoding step. By doing so, both branches can learn the knowledge from each other and improve itself simultaneously. In order to solve the problem of label imbalance, we introduce the focal loss to enhance the training of the hard examples. We use multi-scale attention to improve the problem of variant scale of handwritten math symbols. Compared with previous methods, our method achieves state-of-the-art accuracy of 57.46 % on CROHME 2014, 53.76 % on CROHME 2016 and 53.63 % on CROHME 2019 by only using the official training dataset and without data augment. Our model could achieve significant improvements without any additional parameters and computational cost during validation.

Introduction

Handwritten mathematical expression recognition (HMER) is an important research direction in handwritten recognition field, and it has wide applications, such as intelligent education, human-computer interaction and academic paper writing auxiliary tools. Traditional recognition methods generate the LaTeX sequence from an input image based on specially designed grammars rules (Lavirotte and Pottier 1998; Chan and Yeung 2001; MacLean and Labahn 2013). They do not fit the mathematical expression with complex structure, because it is difficult for them to capture the intrinsic positional relationship between symbols in HME.

Recently, the attention based encoder-decoder models have attracted much attention due to its excellent performance in the filed of machine translation (Cho et al. 2014), speech recognition (Bahdanau et al. 2016) and character recognition (Zhang et al. 2017b). Deng (Deng et al. 2017) first introduces encoder-decoder frame to convert HME image to LaTeX based on attention without requiring a textural or visual grammar. Le (Le and Nakagawa 2017) proposes a local and global distortion models for generating large data for training. Zhang (Zhang et al. 2017a) proposes coverage attention with considering the past attention information when producing the sequence word by word and also propose multi-scale DenseNet as the encoder to improve the performance (Zhang et al. 2018). Moreover, Zhang (Zhang et al. 2020) proposes a tree-structured decoder aiming at generating a tree-structured to handle complex formula and can improve the model's ability to recognize structures. However, its performance may be affected by the quality of the grammar structure tree and difficult converge. Zhao (Zhao et al. 2021) first attempts to use end-to-end transformer decoder and performs bidirectional language modeling to solve HMER task. Although it can solve the problem of gradient disappearance caused by long dependence compared with classic sequence model (RNN), the ability of characters and grammatical structure recognition is insufficient in formula recognition.

These methods have achieved outstanding performance, compared with grammar-based methods, but there still are some limitations. First, the existing methods suffer from the lack of coverage problem (Zhang et al. 2017a) that consists of two types: over-parsing and under-parsing, which indicate some words in the expression are repeatedly translated multiple times, or are not translated. Those methods only use a single-scale attention mechanism on the context feature, causing an under-parsing problem. Second, these method all face a large class imbalance during training. For example, in dataset of CHROME 2014, the label number of character "{" is five of thousands of times the number of the character '/exist'. The minority class is harder to predict because there are few examples of this class and this means it is more challenging for the model to learn the characteristics of examples from this hard class. This imbalance will cause the inefficient training as easy example contribute no useful learning signal and affect the recognition accuracy of minority class. The classification error in the training.

The mutual learning method in knowledge distillation has recently received widespread attention, and has been widely used in classification and segmentation tasks. In mutual learning, two identical or different models learn from each

other through the final logits output and promote together.

Motivated by these observations, we propose a Cooperative Attentive Translation Network (CATNet) for HMER task with an Encode-Multi-Decoder architecture, as illustrated in Fig. 1. This structure consists of one encoder uses to extract detail feature from the input image and multiple parallel decoder branches in which each shares the same structure and attention mechanism except the weight initialization method.,The two decoders separately predict the result at each time step and generate one sequence at end. In this frame, We first propose to introduce a mutual learning framework (Zhang et al. 2018) to jointly train two branches by learning each other at each decoder step. In each step of decoding, each branch will calculate an attention score on the feature map to indicate the position of the translation at the moment. The two branches learn the position of attention from each other to help improve the accuracy of recognition. By doing so, both branches can fully absorb the knowledge from each other and thus could be improved simultaneously. This contributes to the solve the problem of over-parsing and under-parsing and effectively alleviates the problem of lacking coverage. In order to allow the two branches to learn from each other, we use KL divergence to narrow the distance between the distribution of the two prediction results. It is necessary to mention that although we use the multi branches for training, we could approximate the multi-branch network with only one major branch during inference and our method . Further, to avoid the problem of label imbalance, we leverage the tradition method to help the model pay more attention on the hard example, Different from (Zhang et al. 2018), we computer the past attention probabilities on multi-scale due to the variety of the character size in the ME on one feature map. Our contribution are summarized as follows:

(1) We firstly introduce mutual learning for mathematical formula recognition filed, and propose a new novel training strategy for the image-sequence task, where multi-branch decoder can learn from each other at each decoding step.

(2) We propose a multi-scale attention on the past attention map to better identify the variant scale symbols. A focal loss is introduced into HEMR to solve the label imbalance problem.

(3)Our method is applicable to existing decoders based on GRU, LSTM, and transformer structures, and can effectively improve their performance without increasing parameters.

(4) From the comprehensive experiments, we demonstrate the advantage of the new learning strategy for the encoder-decoder frame. And the performance of our model highly surpasses the state-of-the-art on CROHME 2014, 2016, 2019.

Related works

Methods of HMER

Handwritten Mathematical Expression Recognition (HMER) aims to automatically generate the LaTex from given an input expression image. Traditional methods requires special designed grammars rules to solve the two-dimensional structure of expression: graph grammars (Lavirotte and Pottier 1998), definite clause grammars (Chan and Yeung 2001), relational grammars (MacLean and Labahn 2013) and probability model based grammars (Awal, Mouchere, and Viard-Gaudin 2014; MacLean and Labahn 2015; Álvaro, Sánchez, and Benedí 2016).

In recent years, based on the wide application and performance of the encoder-decoder frame on machine translation (Luong et al. 2014) and parsing (Vinyals et al. 2015), many approaches propose to use encoder-decoder frame to solve this image to sequence task. Deng (Deng et al. 2017) firstly introduces the encoder-decoder frame into HEMR, in which a convolution neural network (CNN) and a RNN (Kawakami 2008) are used as the encoder to capture the sequential order information, and a GRU (Chung et al. 2014) based on attention mechanism is introduced to achieve the decoding from the high-level feature to LaTeX symbols. Many approaches typically improve the encoder with the stronger convolutional networks to strengthen feature extraction: Full Connected Network (Zhang et al. 2017a), densely connected network DenseNet (Zhang, Du, and Dai 2018; Le, Indurkhya, and Nakagawa 2019), ResNet-18 (Li et al. 2020). However, one of the major drawbacks of these approaches is that they fail to consider the attention relation of feature map. During the decoding, It is found that better translation alignment is achieved after adding attention in the decoding process. Attention mechanism is added into te decoder to achieve better translation alignment, such as coverage attention (Zhang et al. 2017a; Zhang, Du, and Dai 2018) that consider the all past attention probabilities to improve the problem of lacking coverage. Besides, some methods establish additional expression dataset to improve the performance. Anh Duc (Le, Indurkhya, and Nakagawa 2019) proposes pattern generation strategies to generate different shape and structural handwritten mathematical expression variations. Li (Li et al. 2020) improving the unstable scale on mathematical expression by random scale augmentation but keeping the original aspect ratio. And Truong (Truong et al. 2020) achieves weak-supervised training by adding a symbol classifier to classify the symbol in high-level feature map. Zhao (Zhao et al. 2021) uses a transformer-based decoder to replace classic RNN-based ones and proposes image positional encoding to accurately assign the attention to different regions of input image. We follow the encoder-decoder frame and improve the model with a refreshing training strategy with mutual learning.

Mutual learning

Mutual learning refers to the process of mutual learning of the same or different models, which has been widely exploited for improving the modeling capacity. DML (Zhang et al. 2018) first propose the concept of mutual learning in the knowledge distill domain, which improve the generalisation ability of a network by training collaboratively with a cohort of other networks with KL divergence to match the two network's predictions. Co-distillation (Anil et al. 2018) force each network maintain their diversity with distillation loss. ONE (Lan, Zhu, and Gong 2018) train a multi-branch network and ensemble predictions from these branch as soft label to guide each branch network. CLNN (Song and Chai

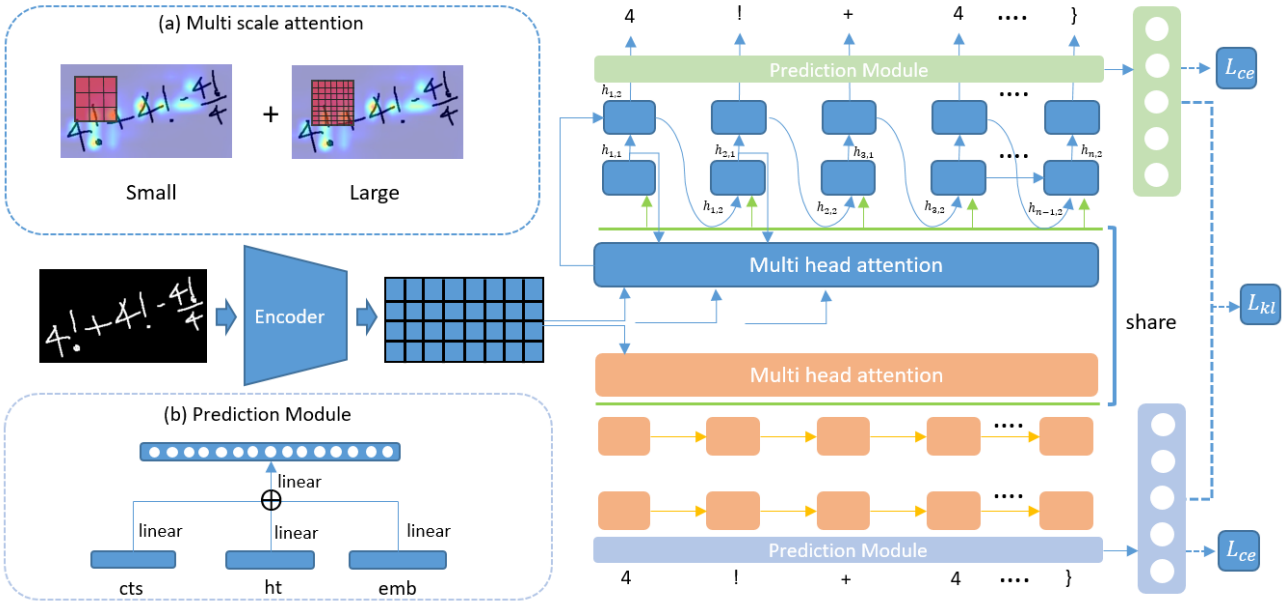

Figure 1: Complete architecture of our proposed model. The numbers of the decoder can be added for mutual learning. During the decoding in each time step, two branches are trained to minimize the distance between the predicted probability of two branches.

2018) design a hierarchical multiple branch and propose to scale the gradient accordingly. KDCL (Guo et al. 2020) propose multiple methods to ensemble soft target from multi-network and then supervise the learning of each network. This only applies in this simple classification task situation where each single network learn each other, and each network accepts input images and output predictions. We propose a new mutual learning method that aim at encoder-decoder networks, in which two branch decoder respectively the high-level feature from encoder at the same time.

Method

The architecture of our end-to-end hand mathematical expression recognition is shown in Fig. 1. Give an image I containing the mathematical expression, our model can output LaTex sequence Y. It has three modules: an encoder to extract feature from the input image, two Decoders to jointly generate the target LaTeX probability while learning from the other. During training, we except each decoder branch to fully learn knowledge not only from its label but also the action prediction distributions of the other branch. Need to mention that our framework is also suitable for training of more than two decoders, as show in the experiment part.

Architecture

Encoder We use DenseNet as the encoder to extract the feature from an input image similar to zhang(Zhang, Du, and Dai 2018). The output of it is a three-dimensional feature map F of $C \times H \times W$, where C,H and W respectively denote channel, height and width. From this, we consider the output as content information a at each position of in the image, $\mathbf{a} = \{a_1, ..., a_L\}, a_L \in \mathcal{R}^C, L = H \times W$. $\mathbf{a}$.

Y-shaped decoder Instead of using one decoder same as the traditional method (Zhang et al. 2017a; Zhang, Du, and Dai 2018), we propose dual stream decoder.The decoder generates a corresponding $LaTeX$ sequence of the input handwritten mathematical expression. The probability of each predicted symbol is computed by the following equation:

$$p(y_t|y_{t-1}, X) = W_o max(W_y(E(y_{y-1})) + W_h h_t + W_t c_t), \quad (1)$$

where h_t, y_{t-1} and c_t denote the current state, previous output and context vector calculated by attention on context feature at $t-1$ or t steps. And the details of c_t will be shown in the next chapter. W_o, W_y, W_s and W_c are trainable weight matrix that are $\in \mathcal{R}^{d \times K}$, $\in \mathcal{R}^{D \times d}$, $\in \mathcal{R}^{d \times d}$, $\in \mathcal{R}^{d \times d}$. And d=256, D=684, K=111. E is a linear layer that embeds input token in a continuous vector. max denotes a maxout activation function. The hidden representations $\{h_1, h_2, .., ht, .., h_n\}$ are produced by

$$h_t = f(h_{t-1}, E(y_{y-1})), \quad (2)$$

Where f denotes the GRU model similar with (Zhang et al. 2017a).

Multi-scale Attention The accuracy of model is related on being able to precisely track the current location on the image when the decoder generate a corresponding LaTeX sequence at t step. The current location is obtained through attention from context vector f, hidden vectorh_{t-1} and past attention probabilities β_t (Zhang et al. 2017a; Zhang, Du,

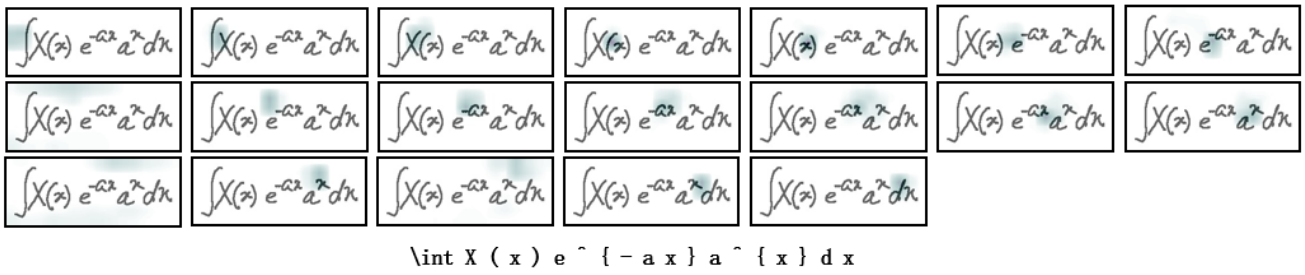

Figure 2: Attention visualization of the recognition process translating HME to LaTeX sequence at each step.

and Dai 2018; Li et al. 2020). The position of the attention at the $t-1$ step on the past attention probabilities $beta_t$ directly determines which position should be decoded at the next step t. For Example, in a fractional formula, if the numerator is decoded at the previous moment, the attention of the next moment must be on the denominator. Therefore, it is extremely important to accurately locate the relative position of the characters at each moment. Previous methods always use large-scale convolution, we use large and small convolution to pay attention to large and small characters at the same time. In the following equation, U_a and U_b denote the convolution operation of different kernel size, $e.g.$ 3*3 and 11*11. We need to mention that our multi-scale attention is different from the Multi-scale attention by (Zhang et al. 2018), the latter generate low-resolution features and high-resolution features and respectively.

$$\beta_t = \sum_{l}^{t-1} \alpha_t, \tag{3}$$

where α_t denotes the attention score at step t, β_t represents the sum of all past attention probabilities, which is initialized as zero vector.

$$e_t = v_a^T \tanh(W_a h_{t-1} + U_f \mathcal{F} + U_a \beta_t + U_b \beta_t), \tag{4}$$

$$\alpha_t = \frac{exp(e_t)}{\sum_{k=1}^{L} e_{tk}}, \tag{5}$$

$$c_t = \sum_{i}^{L} \alpha_t \mathbf{a}_i, \tag{6}$$

where F denote the high-level feature extracted from encoder. c_t shows the attention on different hidden states h_t, which is computed as a weighted sum of the feature map F with the attention mechanism in the time step t.

Loss Function

In this work, we handle the multi-class classification task. Assuming that one sample I of L length from M classes, the logits of the l-th symbol of sequence forward by the decoder network is defined as $z_l = \{p_1, p_2, ..., p_m\}$. We denote this corresponding one-hot ground-truth label as $Y_l = \{y1, y2, ..., y_m\}$ with $y_i \in \{1, 2, ..., M\}$. The probability of class m for the l-th symbol is computed as:

$$\sigma(z_l) = \frac{e^{z_l^i}}{\sum_{j=1}^{m} e^{z_l^j}} \tag{7}$$

For multi-class classification, the cross-entropy loss is defined as:

$$L_{ce} = \sum_{j=l}^{L} \sum_{i=1}^{m} -Y_l^{(i)} log(\sigma(z_j^i)), \tag{8}$$

To solve the label imbalance, we use the dual focal loss to replace the cross entropy (CE) loss for classification (Hossain, Paplinski, and Betts 2019):

$$L_{fl} = -\sum_{j=1}^{L} \sum_{i=1}^{m} log(1 - (z_j^i - Y_j^i)^2), \tag{9}$$

$$\frac{\partial(L_{fl})}{\partial Y_j^i} = -\frac{2(z_j^i - Y_j^i)}{1 - (z_j^i - Y_j^i)^2}, \tag{10}$$

It pays more attention on hard examples when training in addition to the easy examples, which directly use the squared difference between the ground truth Y_l and predicted probability z_l.

Mutual learning In order to learn the action prediction distribution from the other branch, we introduce Kullback-Leibler (KL) loss to quantify the action prediction distribution divergence between the two branch. In the training process, we use softened probability of the model generalization to provide more information. We define the output logits from the k-th branch network Θ_k as $z_{l,k}$, therefore, the softened probability is defined as

$$\sigma(z_{l,k}, T) = \frac{e^{z_{l,k}^i / T}}{\sum_{i}^{m} e^{z_{l,k}^i / T}} \tag{11}$$

The k-th learn the knowledge from the other K-1 branches, and KL loss of the soft output of this branch is defined as below:

$$L_{kl}^k = \frac{T^2}{n} \sum_{q=1, q \neq k}^{K} \sum_{i=1}^{n} \sigma(z_{l,k}, T) log \frac{\sigma(z_{l,k}, T)}{\sigma(z_{l,q}, T)}, \tag{12}$$

where T is the temperature parameter (T=4), p and q is the softmax output from the soften probability distribution generated by two branches.

Dataset	methods	ExpRate	≤1 error	≤2 error
$\mathcal{I}$	WAP	46.55	61.16	65.21
	DWAP-TD	49.10	64.20	67.8
	DWAP	50.60	68.05	71.56
	DWAP-MSA	52.80	68.10	72.00
	WS WAP	53.65	-	-
	BTTR	53.96	66.02	70.28
	Ours-1	57.40	72.92	80.58
	Ours-2	56.69	72.11	81.14
	Ours-E	56.95	72.92	81.03
	Ours-1-LM	**58.62**	**74.34**	**81.74**
$\mathcal{II}$	WAP	44.55	57.10	61.55
	DWAP-TD	48.50	62.30	65.30
	DWAP	47.43	60.21	63.35
	DWAP-MSA	52.80	68.10	72.00
	WS WAP	51.96	64.34	70.10
	BTTR	52.31	63.90	68.61
	Ours-1	53.53	69.31	78.38
	Ours-2	53.62	69.92	78.73
	Ours-E	53.69	69.83	78.81
	Ours-1-LM	**55.54**	**71.40**	**79.95**
$\mathcal{III}$	DWAP	41.7	55.50	59.30
	DWAP-TD	51.40	66.10	69.10
	BTTR	52.96	65.97	69.14
	Ours-1	53.54	68.72	76.75
	Ours-2	53.96	69.47	77.31
	Ours-E	54.05	69.31	76.90
	Ours-1-LM	**54.96**	**69.64**	**77.65**

Table 1: Comparison with the prior works on the three different dataset: CROHME 2014 ($\mathcal{I}$), CROHME 2016 ($\mathcal{II}$) and CROHME 2019 ($\mathcal{III}$). Ours-1 Ours-2, Ours-E and Ours-LM denote the results from sub-branch 1, the results from sub-branch 2, the ensemble results of the two branches and the results of using language model when decoding using sub-branch 1. We need to point that we chose the model that performed best in CROHME 2014 test data and then test it in the other two test set($\mathcal{II}$ and $\mathcal{III}$).

The overall loss function is as following:

$$L = \sum_{k=1}^{K} (L_{ce}^{k} + \alpha L_{kl}^{k}) \tag{13}$$

where α is a tunable parameter to balance the cross-entropy and KL divergence.

Experiment

Dataset and training details

We train the network using CROHME 2014 competition dataset (8894) which contains 101 classes of math symbols and test our model on three public datasets CROHME 2014 (986), CROHME 2016 (1147), CROHME 2019 (1199). All the models are trained on one GPU V100 with batch size 40.

We use the Adadelta optimizer and set the weight decay to 0.0001, the max norm of the gradients of gradient clipping to 100 (Zeiler 2012). The learning rate starts with 1 and use learning rate reduction strategy. If the model is trained continuously for 15 iterations, WER (Word Error Rate) does not decrease in validation, and the learning rate will be halved. And the model will early stop if the learning rate drops 10 times. We need to note that different branches in our model are set to use different weight initialization methods. In the loss function, α is set as 5000. We use four metrics to evaluate our expression recognition model at expression level and word level. (1) expression level: ExpRate (%),$\leq$ 1 error (%) and $\leq$ 2 (%) error respectively mean expression recognition accuracy when zero to two structural or symbol errors can be tolerated; (2) word level: WER (Klakow and Peters 2002) is a metric that can evaluate the errors in word level, such as, substitutions, deletions and insertions.

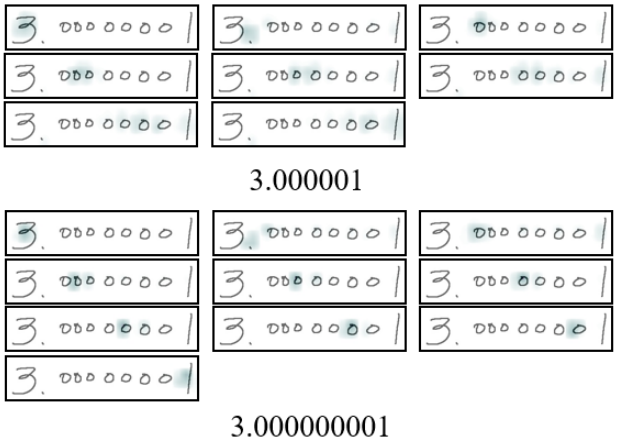

Figure 3: Incorrect and correct attention processes of two examples recognized by baseline (top) and our model (bottom). Our method helps to focus on the correct position in every step of decoding.

Comparison with prior work

To evaluate the our method, we compared it with the previous state-of-the-art methods in three test dataset: CROHME 2014 (986), CROHME 2016 (1147), CROHME 2019 (1199) in term of the recognition accuracy, including WAP (Zhang et al. 2017a), WAP with DenseNet as encoder (DWAP) (Zhang et al. 2018), WAP with DenseNet as encoder and tree decoder (DWAP-TD) (Zhang et al. 2020) Weakly supervised WAP (WS WAP) (Truong et al. 2020) and bidirectionally trained transformer (BTTR) (Zhao et al. 2021). Table 1 shows the recognition results of different methods.

We can see that our proposed method greatly improve the performance of Dense WAP (baseline) (Zhang et al. 2018). The recognition accuracy of our method outperforms baseline (DWAP) by 14.77 %, 13.69 % and 11.33 % in three test dataset, respectively. DWAP uses DenseNet to extract feature from input image, which improve the model 2.96 % in recognition accuracy compared with WAP. DWAP-TD with tree decoder enhance the decoding ability of handling complex formulas. Due to its complex decoding rules, although

Methods	ExpRate	$\leq$1 error	$\leq$2 error	WER
Baseline	50.60	68.05	71.56	13.12
+M-branch	52.74	68.86	77.69	11.73
+KL	54.56	70.49	78.30	11.15
+Focal loss	56.29	73.02	81.14	10.27
+M-scale	56.55	72.82	80.22	10.27
+Embedding	57.40	72.92	80.58	10.22
+LM	58.64	74.34	81.74	9.73

Table 2: Ablation study. We evaluate the performance of each experiments with ExpRate (%), $\leq$1 error (%), $\leq$2 error (%) and WER (%) on CROHME 2014 test dataset. "Baseline" denotes the original Baseline model based on encoder-decoder architecture from WAP (Zhang et al. 2017a). "+" denotes to append the current part to the previous system.

Methods	del	ins	str	total	sub	total
Baseline	147	83	28	258	229	487
+Two-branchs	135	82	24	241	225	466
+KL	142	54	23	217	229	448
+Focal	140	46	24	210	221	431
+M+E	125	51	20	193	224	420
+LM	120	55	20	193	213	408

Table 3: Statistics of the four types of recognition errors on CROHME 2014 test dataset. 'del', 'ins', 'str' and 'sub' are respectively deletion errors, insertion errors, structure errors and substitution errors. 'del', 'ins' and 'str' this three kind of errors are mainly caused by the under-parsing or over-parsing during the decoding. The result illustrates that our model mainly solve the problem of translation misalignment.

it can make the decoding result conform to the grammatical rules to a certain extent, it cannot solve the long sequence and complex structure formulas due to the lack of converge. BTTR can indeed solve the problem of long sequence decoding errors to improve the overall recognition accuracy, but from the results of ≤ 1 error and ≤ 2 error, we can see it cannot fundamentally solve the character decoding error as the word error rate has not improved. However, in terms of ≤ 1 and ≤ 2 error, our model outperforms the best model (BTTR) by about 7 % in $\leq$ error and 10 % in $\leq$ error. Therefore, our proposed method can fundamentally improve the problem of word decoding errors. We also show the ensemble result of two branches, which is obtained by averaging their prediction probabilities at each decoding step.

The fusion of the two branches does not improve the final accuracy as show in the Ours-e row of Table 1. In the process of mutual learning, each branch actually learns the ensemble result of the two branches, and finally each branch has the same ability as the ensemble method. This phenomenon is also in agreement with previous work.

Attention Visualization

Fig. 2 shows an example of the recognition process by our proposed method. The dark blue color denotes a high attention probabilities on the context feature at current decoding step, and there is greater probability of decoding the character at this position. We can see that the attention on basic symbols ("X", "e",) is dark blue while the attention on implicit relation symbols (like "{", "}", "") is lighter. This is because the implicit symbols "" is automatically parsed when encountering a superscript relationship.

Ablation Study

We conduct ablation study to investigate the contribution of our proposed components in the network, which is shown in Table 2. The baseline model is a basic encoder-decoder architecture (DWAP) (Zhang et al. 2018), which achieves ExpRate 50.60.

We make an incremental experiment to verify the importance of the proposed module from the baseline to the final model (Our model). "+Two-branches" is the case that uses two decoder model to translate the feature from the encoder. As expected, Adding the multi-branch block in this basic model highly increases recognition ability of the model by the mutual learning, and the ExpRate highly increases from 50.60 to 52.74.This illustrate that our multi-branch design bring some positive regularisation effect by the jointly learning . From the "+KL", we find the KL divergence can better promote the mutual learning between the two decoder branches improving the ability of recognition. This result confirm that our method enable the sufficient join learning obtaining rich information from two branches. This assistant supervision brings about 4.16% increment for the other branch. "+Focal loss" is the case using Focal loss instead of the classic cross entropy loss as the classification loss. From the $\leq$ 1 error and $\leq$ 2 error, the recognition accuracy of a single character has been greatly improved and achieve the highest accuracy 81.14 %. This attributes to the introduced focal loss that solves the problem of symbol label imbalance and it can better recognize difficult characters. From "+M-scale", the Multi-scale attention benefits the structure recognition because of the more attention on small symbols while single character recognition ability is reduced. Further, Adding the share of embedding layer ('+Embedding') between two branches leads to better performance. The plausible reason is that the same embedding input increase the stability and consistency of decoding between two branch. '+LM' denotes using language model. We use the the training dataset to train the language model. At each decoding step, the probability of the language model predicting the next character and the probability predicted by this model are used to predict the next character together. This further improve the performance of recognition by over 1.24 %. In brief, each component in our method brings improvements to the recognition and they are complementary to each other.

Advantages of Our Approach

In this section, we try to shed some light on why and how our model is better than the baselines. What can our model can improve the baselines model during the decoding?

More Accurate Attention We use quantitative and qualitative experiments to verify the attention improvement of

the proposed method. Fig. 3 shows two typical examples respectively generated by the baseline model and our proposed model. In the first one, we can see that the attention is inaccurate when translating the fifth character '0', leading to the under-parsing problem. In the second case, attention is accurately focused on the previous character in each step of decoding, so that the next decoding is correct, so our method can promote concentration to a certain extent and help the coverage of attention.

In order to quantitatively analyze the problem and more accurately determine what problems our method can solve, we divide the types of decoding errors into 4 categories: structure error caused by can not translate the implicit structural relationship such as superscript "", subscript ""; delete error caused by missing several characters during translating when it is difficult to distinguish the overlapped or small characters, such as the "3.000001" in Fig. 3; insert error happen when a few more characters were incorrectly translated; substitution error happens when they are not written clearly, such as Upper and lower case letters, similar characters "z" and "2". Table 3 represents the numbers of each error on different methods

More powerful generality Does our method have the same effect on other types of decoders? Will different Encoder affect the results of multiple decoders? To demonstrate the effectiveness of our method, we conduct experiment on DWAP with different Main-stream backbone architectures. In each experiment, we compare the result before and after using our method. As shown in Table. 4, for CNN-based backbones (Xception, ResNet18, and ResNet34), our method can boost baseline by over 3.5%. Therefore, our model is compatible with different backbone network.

Further, in order to inspect the generality of our proposed framework on different types of decoders (GRU (DWAP) (Zhang et al. 2017a), LSTM (DWAP-LSTM) and transformer (BTTR) (Zhao et al. 2021)). In this experiment, we only replace GRU with LSTM in DWAP to form DWAP-LSTM. As shown in Table 4. Our method improves GRU, LSTM and transformer by 6.8 %, 7.31%, and 2.36 %, respectively. BTTR-Dual only improve the baseline by 3.26 % may because the transformer is not easy to training. Therefore, our method is not only compatible with CNN-based backbones, but also with the different decoders.

Comparison with other mutual learning methods

In this section, we discuss on the similarity and difference of the proposed method with other mutual learning methods DML (Zhang et al. 2018), ONE (Lan, Zhu, and Gong 2018), KDCL (Guo et al. 2020) and PCL (Wu and Gong 2021). The proposed method is more similar to ONE. The main difference from ONE is multiple branch variants constructed by our proposed method, which are independent decoders not from the higher layers. This design contributes the coverage of attention during the process of recognition. And the all previous mutual learning methods always use entire complete model in term of the application of the classification and segment tasks. Traditional mutual learning methods is that one network imitate the other network in the training process through the cross-entropy loss between each pair of network. However, for the encoder-decoder architecture with attention mechanism, does this training strategy also fit this model?

Methods	ExpRate	$\leq$1 error	$\leq$2 error
WAP-M2	40.61		
WAP-M2-Dual	46.29		
WAP-X	43.05		
WAP-X-Dual	45.69		
DWAP	50.60	68.05	71.56
DWAP-Dual	57.40	70.49	78.30
DWAP-LSTM	49.64		
DWAP-LSTM-Dual	56.95		
BTTR	48.13		
BTTR - Dual	50.46		

Table 4: The performance of different methods equipped with our training strategy. BTTR and DWAP composited with our method is more effective than its own vision.

Methods	ExpRate	$\leq$1 error	$\leq$2 error
Baseline	50.60	68.05	71.56
Baseline+KD			
Baseline+DML	55.23		
Baseline+DML w L_2	53.20		
Baseline+KDCL-naive			
Ours(One+Two)	57.40	72.92	80.58

Table 5: Comparison with different mutual learning strategy in the HMER task. All model use same training strategy except for the mutual learning methods. DML, KDCL are the method on (Zhang et al. 2018) and (Guo et al. 2020). 'L2' indicates using L2 loss on the feature map between the feature map from two encoder model in DML training process. *+*' denotes the model consists of how many encoders and decoders and other setting are same, $e.g.$ 'One+Two' using one encoder and two decoder in the network.

We verify the effect of those methods on HMER task, and the results are not ideal showed in Table 5. The reason may be caused by the the complexity of the HMER in which the decoder conducts attention on the context feature, and if the context feature are not same, may exacerbate the inaccuracy of the attention on the symbols. Our method can avoid this problem by directly using single feature from one encoder. Experimental results demonstrate that the improvement from our learning mechanism can be better suitable for this kind of task.

Further Analysis and Discussion

Why Using one Encoder instead of two? In this section, we discuss the model with two encoders and one decoder, the result is shown in Table 6. Intuitively, a decoder learns features and performs attention calculations on two different feature maps at the same time, which leads to unstable decoding. From the experimental result, the performance is bad than the baseline may because of inconsistent learning. We also evaluate the efficiency of our method by controlling

Methods	ExpRate	$\leq$1 error	$\leq$2 error
Baseline	50.60	68.05	71.56
Two+One	50.46		
Ours(One+Two)	57.40	72.92	80.58
One+Three	56.39	71.71	79.31
One+Four	55.68	72.01	79.92

Table 6: Effect of the number of the encoders and decodes in our model. Baseline model consists of one encoder and one decoder. *+*' denotes the model consists of how many encoders and decoders and other setting are same, $e.g.$ 'One+Two' uses one encoder and two decoder in the network.

Methods	CROHME 2014				
	[1,10]	[11,20]	[21,30]	[31,]	[41,$\propto$]
Baseline	36.07	45.96	52.83	56.91	73.17
WAP-TD					
BTTR					
Ours-sub1	26.43	39.19	46.22	45.52	65.85

Table 7: Percentages of errors with respect to the different length of ME LaTex sequences on CROHME 2014.

the number of the decoders (N = 2, 3, 4). Note that the accuracy decrease as the number of the decoder increase. our method provides an effective and efficient alternative to improve the model performance rather than simply increasing the depth or width of the network.

Performance of the Size of Expression Further, in order to explore the ability of our method to decode sequences of different lengths, we split test datasets into different group according to the length of their ME LaTex sequences and then compare the model performance at each group. Intuitively, the longer the expression, the more difficult it is to translate. We expected poor perform due to the limited ability to parse the long sequence. From Table **??**, we can observe that our model outperform or achieve comparable performance than this corresponding baseline and other methods respectively.

Conclusion

We propose a simple training strategy to improve the performance of HMER by training a multi-branch decoder to help learn from each other. This approach is suitable to the encoder-decoder network and the performance is better than the traditional mutual learning methods, such as DML. We also show that the proposed multi-head attention is promising to improve the problem of converge.

References

Álvaro, F.; Sánchez, J.-A.; and Benedí, J.-M. 2016. An integrated grammar-based approach for mathematical expression recognition. *Pattern Recognition* 51: 135–147.

Anil, R.; Pereyra, G.; Passos, A.; Ormandi, R.; Dahl, G. E.; and Hinton, G. E. 2018. Large scale distributed neural network training through online distillation. *arXiv preprint arXiv:1804.03235* .

Awal, A.-M.; Mouchere, H.; and Viard-Gaudin, C. 2014. A global learning approach for an online handwritten mathematical expression recognition system. *Pattern Recognition Letters* 35: 68–77.

Bahdanau, D.; Chorowski, J.; Serdyuk, D.; Brakel, P.; and Bengio, Y. 2016. End-to-end attention-based large vocabulary speech recognition. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 4945–4949. IEEE.

Chan, K.-F.; and Yeung, D.-Y. 2001. Error detection, error correction and performance evaluation in on-line mathematical expression recognition. *Pattern Recognition* 34(8): 1671–1684.

Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* .

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* .

Deng, Y.; Kanervisto, A.; Ling, J.; and Rush, A. M. 2017. Image-to-markup generation with coarse-to-fine attention. In *International Conference on Machine Learning*, 980–989. PMLR.

Guo, Q.; Wang, X.; Wu, Y.; Yu, Z.; Liang, D.; Hu, X.; and Luo, P. 2020. Online knowledge distillation via collaborative learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11020–11029.

Hossain, M. S.; Paplinski, A. P.; and Betts, J. M. 2019. Adaptive Class Weight based Dual Focal Loss for Improved Semantic Segmentation. *arXiv preprint arXiv:1909.11932* .

Kawakami, K. 2008. Supervised sequence labelling with recurrent neural networks. *Ph. D. thesis* .

Klakow, D.; and Peters, J. 2002. Testing the correlation of word error rate and perplexity. *Speech Communication* 38(1-2): 19–28.

Lan, X.; Zhu, X.; and Gong, S. 2018. Knowledge distillation by on-the-fly native ensemble. *arXiv preprint arXiv:1806.04606* .

Lavirotte, S.; and Pottier, L. 1998. Mathematical formula recognition using graph grammar. In *Document Recognition V*, volume 3305, 44–52. International Society for Optics and Photonics.

Le, A. D.; Indurkhya, B.; and Nakagawa, M. 2019. Pattern generation strategies for improving recognition of handwritten mathematical expressions. *Pattern Recognition Letters* 128: 255–262.

Le, A. D.; and Nakagawa, M. 2017. Training an end-to-end system for handwritten mathematical expression recognition by generated patterns. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, 1056–1061. IEEE.

Li, Z.; Jin, L.; Lai, S.; and Zhu, Y. 2020. Improving attention-based handwritten mathematical expression recognition with scale augmentation and drop attention. In *2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 175–180. IEEE.

Luong, M.-T.; Sutskever, I.; Le, Q. V.; Vinyals, O.; and Zaremba, W. 2014. Addressing the rare word problem in neural machine translation. *arXiv preprint arXiv:1410.8206* .

MacLean, S.; and Labahn, G. 2013. A new approach for recognizing handwritten mathematics using relational grammars and fuzzy sets. *International Journal on Document Analysis and Recognition (IJDAR)* 16(2): 139–163.

MacLean, S.; and Labahn, G. 2015. A Bayesian model for recognizing handwritten mathematical expressions. *Pattern Recognition* 48(8): 2433–2445.

Song, G.; and Chai, W. 2018. Collaborative learning for deep neural networks. *arXiv preprint arXiv:1805.11761* .

Truong, T.-N.; Nguyen, C. T.; Phan, K. M.; and Nakagawa, M. 2020. 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)*, 181–186. IEEE.

Vinyals, O.; Kaiser, Ł.; Koo, T.; Petrov, S.; Sutskever, I.; and Hinton, G. 2015. Grammar as a foreign language. *Advances in neural information processing systems* 28: 2773–2781.

Wu, G.; and Gong, S. 2021. Peer collaborative learning for online knowledge distillation. In *AAAI*.

Zeiler, M. D. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701* .

Zhang, J.; Du, J.; and Dai, L. 2018. Multi-scale attention with dense encoder for handwritten mathematical expression recognition. In *2018 24th international conference on pattern recognition (ICPR)*, 2245–2250. IEEE.

Zhang, J.; Du, J.; Yang, Y.; Song, Y.-Z.; Wei, S.; and Dai, L. 2020. A tree-structured decoder for image-to-markup generation. In *International Conference on Machine Learning*, 11076–11085. PMLR.

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

Zhang, J.; Zhu, Y.; Du, J.; and Dai, L. 2017b. RAN: Radical analysis networks for zero-shot learning of Chinese characters. *arXiv preprint arXiv:1711.01889* 1–6.

Zhang, Y.; Xiang, T.; Hospedales, T. M.; and Lu, H. 2018. Deep mutual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4320–4328.

Zhao, W.; Gao, L.; Yan, Z.; Peng, S.; Du, L.; and Zhang, Z. 2021. Handwritten Mathematical Expression Recognition with Bidirectionally Trained Transformer. *arXiv preprint arXiv:2105.02412* .