Structure-aware Mathematical Expression Recognition with Sequence-Level Modeling

Minli Li*†
South China University of Technology
Guangzhou, China
seminli_li@mail.scut.edu.cn

Peilin Zhao*
Tencent AI Lab
Shenzhen, China
masonzhao@tencent.com

Yifan Zhang
National University of Singapore
Singapore
yifan.zhang@u.nus.edu

Shuaicheng Niu
South China University of Technology
Guangzhou, China
sensc@mail.scut.edu.cn

Qingyao Wu*
South China University of Technology
Guangzhou, China
qyw@scut.edu.cn

Mingkui Tan‡
South China University of Technology
Guangzhou, China
mingkuitan@scut.edu.cn

ABSTRACT

Mathematical expression recognition (MER) aims to convert an image of mathematical expressions into a Latex sequence. In practice, the task of MER is challenging, since 1) the images of mathematical expressions often contain complex structure relationships, e.g., fractions, matrixes and subscripts; 2) the generated Latex sequences can be very complex and they have to satisfy strict syntax rules. Existing methods, however, often ignore the complex dependence among image regions, resulting in poor feature representation. In addition, they may fail to capture the rigorous relations among different formula symbols as they consider MER as a common language generation task. To address these issues, we propose a Structure-Aware Sequence-Level (SASL) model for MER. First, to better represent and recognize the visual content of formula images, we propose a structure-aware module to capture the relationship among different symbols. Meanwhile, the sequence-level modeling helps the model to concentrate on the generation of entire sequences. To make the problem feasible, we cast the generation problem into a Markov decision process (MDP) and seek to learn a Latex sequence generating policy. Based on MDP, we learn SASL by maximizing the matching score of each image-sequence pair to obtain the generation policy. Extensive experiments on the IM2LATEX-100K dataset verify the effectiveness and superiority of the proposed method.

CCS CONCEPTS

• **Computing methodologies** → **Sequential decision making**; **Computer vision problems**; **Natural language generation**.

*Equal contribution. This work is done when Minli Li works as an intern in Tencent AI Lab.
†Also with Pazhou Laboratory.
‡Correspnding author. Also with Key Laboratory of Big Data and Intelligent Robot (SCUT), Ministry of Education.

MM '21, October 20–24, 2021, Virtual Event, China

ACM ISBN 978-1-4503-8651-7/21/10…$15.00
https://doi.org/10.1145/3474085.3475578

$$A_i^{\perp} = \left(\delta_{ij} - \frac{\delta_i \delta_j}{\delta^2}\right) A_j.$$

(a) The image of a formula

```
A_i^\perp = \left(\delta_{i j}-
\frac{\delta_i\delta_j}{\delta^2}
\right) A_j.
```

(b) The corresponding Latex sequence

Figure 1: An illustration of the challenge in MER. As shown in Figure 1 (a), a formula may contain subscript, superscript and fraction structures. These structures, located in different regions in images, may have strict and complex relations that are essential for mathematical structure understanding and Latex sequence generation. For example, for $A_i^{\perp}$ in Figure 1(a), rather than simply recognizing characters and outputting "Ai\perp", MER needs to consider the complex relations among "A", "i" and "\perp", and generate the Latex code "A_i^\perp" as in Figure 1 (b). Unfortunately, how to exploit these relationships for MER remains a question.

KEYWORDS

Mathematical Expression Recognition, Structure-aware Module, Sequence-level Modeling

ACM Reference Format:
Minli Li, Peilin Zhao, Yifan Zhang, Shuaicheng Niu, Qingyao Wu, and Mingkui Tan. 2021. Structure-aware Mathematical Expression Recognition with Sequence-Level Modeling. In *Proceedings of the 29th ACM International Conference on Multimedia (MM '21), October 20–24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3474085.3475578

1 INTRODUCTION

Mathematical expression is a fundamental tool to symbolically express problems and theories in mathematics, physics and many other fields [3, 38]. In most scientific and engineering disciplines, mathematical expressions are the essential part. Since mathematical expressions contain special/complex symbols, it is often difficult to input mathematical expressions into computers [23]. To handle this, mathematical expression recognition (MER), i.e., translating math formulas from digital documents into markup languages, has become increasingly important in recent years [11, 20, 37]. Nevertheless, the task of MER is non-trivial due to several challenges.

First, how to represent the formula images is non-trivial, as they often contain complex structure relationships, such as subscripts,

superscripts and nested fractions (see Figure 1). These relationships play an important role in both equation understanding and Latex sequence generation, but are difficult to be modeled. To this end, traditional MER systems often contain an interpretation phase, following the symbol location and symbol recognition phases. The interpretation phase implies a structure analysis of spatial relations among symbols using a human-designed parse tree [3]. However, these methods essentially rely on human-designed rules and may lead to limited relation modeling abilities in practice.

Recently, Deng *et al.* [10] has proposed a deep learning-based MER method that designs a multi-row recurrent neural network (namely Row-Encoder) to model complex structure relationships. After that, several studies [6, 40] are devised based on the Row-Encoder model. However, the Row-Encoder only models the relationship in the row direction, but ignores that the spatial relationship among symbols usually spans along with different directions.

Second, the evaluation of the generated Latex sequences is important for the training of MER models, which, however, is very challenging. Existing deep learning-based MER methods [10, 30] formulate MER as a common natural language generation task and use the LSTM with Maximum Likelihood Estimation (MLE). The maximum likelihood training in common sequence generation means that the model is guided by a token-level evaluation, i.e., maximizing the log-likelihood of each predicted token given the previously observed target sequences. In other words, the MLE training forces the model to directly generate the word as same as the target at the token level without any constraints on the sentence level. However, unlike the natural language which focuses on creativity and diversity at the token level, the mathematical expression tends to be rigorous and contextual. Thus, sequence-level guidance is more practical and even necessary for MER. However, sequence-level training still remains an open question in MER.

To resolve the above challenges, we propose a structure-aware model with a sequence-level modeling, namely SASL. Our model follows a generator-discriminator structure. The generator, including an encoder and a decoder, aims to generate Latex sequences. The encoder consists of a convolutional neural network for feature extraction and an innovative structure-aware attention-based module for structure relationship modeling. Meanwhile, we devise the decoder based on Transformer [31] to decode the extracted features into Latex sequences. To train the generator at the sequence level, inspired by that the sequence generation process is essentially a multi-step decision making process, we propose to model the image-to-Latex process as a Markov Decision Process and solve the problem using Reinforcement Learning.

One key question is how to provide reward signals. Recently, modeling the sequence-level guidance with adversarial training has been verified to be promising in controlled text generation [16] and image captioning [9]. Hence, a discriminator can be regarded as a good evaluator to provide sequence-level feedback for MER model training. Inspired by this, we train a discriminator to provide the sequence-level training reward by distinguishing whether the generated Latex sequence matches the input image well. Moreover, our discriminator provides stepwise evaluation feedbacks, which is different from those RL-based image captioning methods [9, 25] that only provide feedbacks for the whole sequence. In this way, our method leads to more stable policy learning. Extensive experimental results demonstrate the effectiveness and superiority of our proposed method.

Our main contributions are summarized as follows:

(1) We innovatively model the Latex sequence generation process as a Markov Decision Process (MDP), which has not been explored by previous MER methods. Based on the MDP, we solve the problem using reinforcement learning.
(2) We devise a novel discriminator model to provide reward signals in the MDP. By evaluating how well the generated sequences match the input image, the discriminator is able to provide informative reward signals and thus benefits the learning of the generation model.
(3) We propose a structure-aware feature extraction module for MER. By resorting to the self-attention scheme, the proposed structure-aware module is able to model the complex structure relationships among symbols.

2 RELATED WORKS

Mathematical Expression Recognition Mathematical expression recognition (MER) is a subfield of optical character recognition (OCR), which aims to recognize natural language from an image. Traditional OCR usually contains the following stages: symbol segmentation and symbol recognition. Unlike traditional OCR, mathematical expression recognition requires a further structure analysis, such as fractions, matrixes, super-scripts, sub-scripts. Early work [5, 7] proposed to process these three stages separately and model the relationship between symbols via a graph grammar [5, 19]. Moreover, some MER methods [39] used convolutional neural networks to extract features from images, dropping the explicit structure analysis.

In recent years, several studies [6] proposed to follow the encoder-decoder structure with attention and process the above three stages in an end-to-end manner. To be specific, Zhang *et al.* [11] proposed multi-scale attention with a Dense encoder. These models, however, ignore the structure relationship within image regions. To capture the relation information, Deng *et al.* [10] proposed a row encoder, which runs RNNs over each of the rows of CNN features. However, the spatial relationship between symbols usually spans different directions. Therefore, how to model symbol relationships in images still remains an under-explored problem.

Image Captioning Image captioning translates image to a natural language sequence, which is similar to MER. Early approaches [12, 28] usually built a model composed of several independent functional building blocks, including a CNN to extract feature, a language model to generate a set of candidate captions, and a multimodal similarity model to rank those candidate captions. Recent approaches [21, 22, 34] generally followed an encoder-decoder structure [13, 14, 36] with an attention mechanism, in which the encoder extract feature from images and the decoder decodes the extracted feature into a natural language sequence. The attention mechanism [4] was first proposed for neural language translation and now is broadly used to model the relations between image regions and the generated token in image captioning. To learn the captioning model, a common approach is to optimize the cross-entropy

loss, named Maximum Likelihood Estimation (MLE) [12, 21]. Specifically, MLE maximizes the log-likelihood of each predicted token given the previously observed target sequences. This type of optimization is token-level and forces the model to generate tokens that exactly match the ground truth. Some studies [8, 25] propose to adopt some text metrics (e.g., BLEU [24]) to provide a reward signal and train the model at a sequence-level using Reinforcement Learning (e.g., policy gradient [41]). Compared with image captioning tasks, the sequence length in Image-to-Latex tasks is always longer. Directly applying image captioning RL techniques (e.g., [8]) to Image-to-Latex tasks will cause a new delay reward issue, since these methods provide a reward signal after the whole sequence is generated. To address this, we propose to provide reward signals for each generated token rather than the whole sequence.

3 PROBLEM DEFINITION

This paper studies the problem of mathematical expression recognition (MER), which aims to learn a model to translate a formula image into Latex sequences. Formally, let $\{X, Y_{1:T}\}$ denote an image-sequence pair, where X denotes a formula image, $Y_{1:T}$ denotes the ground-truth Latex sequence, T being the length of the sequence[1]. Moreover, the sequence $Y_{1:T}=(y_1, y_2, \ldots, y_T)$ contains a series of tokens $y_t \in \mathcal{Y}(t \in [T])$, where $\mathcal{Y}$ is the vocabulary of candidate tokens. For simplicity, we denote a complete sequence $Y_{1:T}$ by Y. Given a formula image X, MER tries to generate a Latex sequence $\hat{Y}$ to match the ground-truth sequence Y as well as possible. Here, one specific image X may have several different corresponding Y.

MER as a Markov Decision Process. To write the Latex sequences for mathematical expressions, humans generally concentrate on each symbol in the image, and then translate those symbols into Latex tokens sequentially. Such a process is essentially a multi-step sequential decision making process. Inspired by this, we formulate the Image-to-Latex generation process as a Markov Decision Process (MDP). An MDP can be defined as a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$, where $\mathcal{S}$ is a finite set of states, $\mathcal{A}$ is a finite set of actions, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{R}$ is the state transition distribution, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{R}$ is the reward function. Moreover, a policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$ determines an action given the current state. In the context of MER, the MDP is slightly different to the standard one. To be specific, at the generation step t, the state is specific to the pair of the formula image and the historically generated sequence, denoted by $s_t = (X, \hat{Y}_{1:t-1}) \in \mathcal{S}$. Given such a state s_t, a generation policy takes an action $a_t = \pi(s_t) \in \mathcal{A}$ to generate the next token $\hat{y}_t$, and receive a reward $r_t \in \mathcal{R}(s_t, a_t)$. Following that, the next state is reached based on $s_{t+1} = \mathcal{P}(s_t, a_t) = (X, \hat{Y}_{1:t})$, which updates the historically generated sequence.

In this paper, we aim to learn a generator $G_\theta(y_t|X, \hat{Y}_{1:t-1})$ to generate Latex sequences from formula images and maximize the accumulated reward (e.g., the similarity of the generated sequence and the ground-truth one). Here, θ denotes the network parameters of the generator. However, it is not trivial to devise and learn such a generator due to two key questions: (1) how to extract features from images with symbol relationships; (2) how to devise an effective reward signal for training the generator. Existing methods often extract features using a single convolutional neural network, ignoring the complex symbol relationships within image regions. In addition, these methods train the generator by minimizing the negative log-likelihood loss,

$$\begin{aligned} &\min_{\theta} \sum_{i=1}^{N} -\log G_\theta(Y^{(i)}|X^{(i)}) \\ &= -\sum_{i=1}^{N} \log \prod_{t=1}^{T} G_\theta(y_t|X^{(i)}, Y^{(i)}_{1:t-1}). \end{aligned} \tag{1}$$

The above loss function provides a feedback signal at the token level, which focuses on the immediate return and requires the generated sequence exactly to match the only ground truth sequence. However, MER is a multi-step decision making process and Latex sequences tend to be rigorous and contextual. Therefore, this token-level evaluation limits the generation quality of MER in practice. To alleviate the above issues, we propose a structure-aware model with sequence-level modeling for MER.

Algorithm 1 Overall training algorithm

1: Initialize parameters of generator θ and discriminator ϕ.
2: Pretrain generator G using Eqn. (1).
3: Pretrain discriminator D using Eqn. (12) with samples: (1) images in dataset with its corresponding sequence generated by the pre-trained generator; (2) randomly sampled mismatch pairs in the dataset; (3) real data.
4: **for** each training iteration **do**
5: // *Update discriminator*
6: Sample $\{X, Y\}$ from real data;
7: Sample Latex sequences $\hat{Y} \sim G_\theta(Y|X)$;
8: Update D_ϕ using Eqn. (12).
9: // *Update generator*
10: Sample X from real data;
11: Sample Latex sequences $\hat{Y} \sim G_\theta(Y|X)$;
12: Construct dataset $\{X, \hat{Y}\}$ and get reward $D_\phi(X, \hat{Y})$.
13: Update policy G_θ using Eqn. (9).
14: **end for**

4 PROPOSED METHOD

In this paper, we model the Image-to-Latex generation process as a Markov Decision Process (MDP). In an MDP, both the policy network design and the definition of the reward function are significant to policy learning. Since the mathematical expression often contains complex structure relationships, we propose a structure-aware model, which served as a policy network for handling Latex sequence generation. The goal of such a policy network is to extract informative features for mathematical expression recognition and make a multi-step decision based on these extracted features. To learn a well-performed policy, a suitable reward function is of great importance. Since MER focuses on generating a sequence that matches the strict syntax rules, we aim to evaluate the generated sequences at a sequence-level. Considering that modeling the sequence-level guidance with adversarial training is proven to be promising in controlled text generation, we propose to train a discriminator to provide the feedback signal. The overall algorithm is shown in Algorithm 1.

[1]Note that the length of Latex sequences for different formulas can be different.

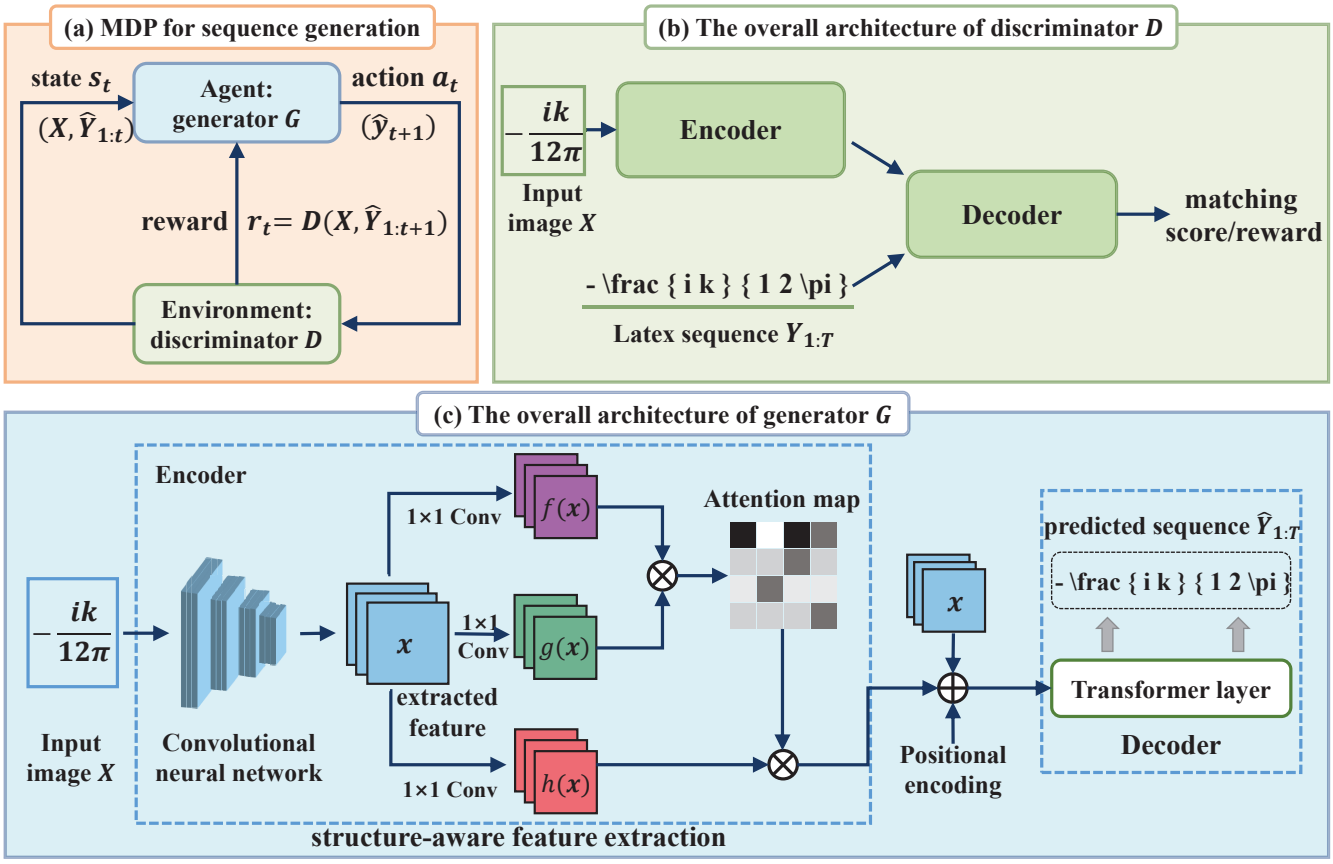


Figure 2: An overview of the proposed method. We formulate sequence generation process as a Markov Decision Process (MDP) and then train the generator G (agent) using reinforcement learning. The reward signals are provided by an additional trained discriminator. The structures of the encoder and decoder in the discriminator are the same as the one in the generator.

4.1 General Architecture

As shown in Figure 2 (a), our proposed method consists of a generator for Latex generation and a discriminator for providing reward training signals.

Generator: As shown in Figure 2 (c), the generator follows the encoder-decoder structure [31]. The encoder extracts a visual feature from images, and then the decoder decodes the extracted feature into Latex sequences. Since the spatial relationship among symbols is significant for MER, it is important to capture such information in visual feature extraction. To this end, we propose a structure-aware module in the encoder for better **structure relationship modeling** (See the next subsection). Moreover, the decoder in our method is identical to the decoder in Transfomer [31]. At each time step, the decoder generates a Latex token based on the input image and the previously generated tokens.

Discriminator: To provide a **sequence-level training guidance**, we devise a discriminator to evaluate the image-sequence pairs. As shown in Figure 2 (b), the discriminator gets the image-sequence pair as the input, and outputs a reward score regarding how well the generated sequence matches the ground-truth one. Please see the next sections for more details.

4.2 Structure Relationship Modeling

In MER, analyzing the structure relationship among image regions is necessary for further Latex sequences generation. Therefore, we involve the encoder with a stucture-aware module to capture the structure relationship. The most commonly used model to extract features from the image is convolutional neural network. The convolution operator only aggregates information in a local neighborhood, leading to inefficiently modeling long-range dependencies. However, long-range dependency is important for an Image-to-Latex system. For example, the left bracket may be far away from the right bracket. To model relationship between symbols, we compute the similarity of the feature vector, which is obtained by the dot product of feature vectors. Similarity of features are commonly used metric to evaluate the relationship, and dot product is widely used to represent the similarity [17, 18]. In this section, we introduce a self-attention mechanism to the image feature extractor. The relationship between spatial regions is represented by the similarity of features enabling our network to efficiently model relationships between spatial regions.

The image features are first extracted from some feature extractors such as convolutional neural networks. Then the extracted

features $\mathbf{x}$ are transformed into two feature spaces using some function $f(\cdot)$ and $g(\cdot)$, i.e., convolution operation or linear transformation. The similarity between location i and location j in the images feature map $\mathbf{x}$ is defined as

$$\alpha_{i,j} = \frac{\exp(s_{ij})}{\sum_{j=1}^{N} \exp(s_{ij})}, \text{ where } s_{ij} = f(\mathbf{x}_i)^T g(\mathbf{x}_j). \tag{2}$$

The attention weights indicate the extent to which the model attends to location j when extracting features for location i. The resulting feature is $\mathbf{O} = (\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_{H\times W})$, where,

$$\mathbf{o}_i = \sum_{j=1}^{H\times W} \alpha_{i,j} h(\mathbf{x}_j). \tag{3}$$

Here, $h(\cdot)$ is some function to transform the original features into another feature space. In our experiments, the feature transformation function $f(\cdot), g(\cdot), h(\cdot)$ are all implemented by 1×1 convolutions. We further multiply the resulting feature map by a scale parameter and add back the original feature map, combining both local and non-local regions. Therefore, the final output is given by,

$$\tilde{\mathbf{x}}_i = \gamma \mathbf{o}_i + \mathbf{x}_i, \tag{4}$$

where γ is a learnable scalar and it is initialized as 0. The learnable parameter γ allows the model to adaptively assign weight to non-local regions.

4.3 Sequence-level Training Guidance

To learn a well-performed MER model, the design of training feedback signals is quite important. Existing MER approaches train the generation model using maximum likelihood estimation, which forces on immediate return and requires the generated sequence completely match the only ground truth sequence. However, MER is a multi-step decision making process and Latex sequences tend to be rigorous and contextual, and thus there is a strong motivation to design a sequence-level training guidance.

To achieve sequence-level training, we propose to learn a Latex sequence generating policy with reinforcement learning based on our predefined MDP. Given some evaluation metric $R(X, Y)$ for evaluating an image-sequence pair (X, Y), we aim to optimize

$$\max_{\theta} J(\theta) = \mathbb{E}_{G_\theta(Y|X)}[R(X, Y)]. \tag{5}$$

To learn the policy G_θ for generating Latex sequences, the necessary component is the definition of the reward signal $R(X, Y)$. In image captioning and NLP, some text evaluation scores (SPICE [2], CIDEr [32] and *etc.*) are used for model evaluation. However, a mathematical expression can have several corresponding Latex sequences, using those text evaluation scores as a reward can lead to an incorrect evaluation.

To alleviate the above issue, we learn a discriminator D_ϕ to evaluate the image-sequence pairs. $D_\phi(X, Y)$ is a probability indicating how likely a sequence is from real sequence data or not. Given an input image X, the discriminator learns to maximize score $D_\phi(X, Y)$ for the real data pair (X, Y) and minimize score $D_\phi(X, \hat{Y})$ for the fake data pair $(X, \hat{Y})$. The generator learns to produce response $\hat{Y}$ to fool the discriminator, i.e., maximizing $D_\phi(X, \hat{Y})$. The optimization problem is

$$\begin{aligned}\max_{\phi} L_\phi = &\mathbb{E}_{(X,Y)\sim P_R(X,Y)}[\log D_\phi(X, Y)] \\ &+ \mathbb{E}_{X\sim P_R(X), \hat{Y}\sim G_\theta(Y|X)}[\log(1 - D_\phi(X, \hat{Y}))],\end{aligned} \tag{6}$$

where $P_R(X)$ and $P_R(X, Y)$ are the probability distribution of X and joint probability distribution of (X, Y) from the training data. We replace the evaluation metric $R(X, Y)$ with the predicted score of discriminator. Since the discriminator evaluates the entire sequences, the rewards except for the last time step are set to zero, which causes the problem of delay reward. To stabilize the policy learning, we aim to make an evaluation at each time step.

We denote $Q(\{X, Y_{1:t-1}\}, y_t)$ as the value of taking action y_t in state $\{X, Y_{1:t-1}\}$ under the policy G_θ, which is the expected return starting from state $\{X, Y_{1:t-1}\}$. Then the state-action value is the expected return of those sequences sharing the same prefix $Y_{1:t}$,

$$Q(\{X, Y_{1:t-1}\}, y_t) = \mathbb{E}_{Z\sim G_\theta(\cdot|X,Y_{1:t})}[D(X, \{Y_{1:t}, Z\})], \tag{7}$$

where Z is a sequence of words generated by the current generator given input X and generated prefix $Y_{1:t}$. Then the optimization problem (5) can be modified as following,

$$\max_{\theta} J(\theta) = \mathbb{E}_{X, y_t\sim G_\theta(\cdot|X,Y_{1:t-1})}[Q(\{X, Y_{1:t-1}\}, y_t)]. \tag{8}$$

The gradient of Eqn. (8) w.r.t.θ is

$$\begin{aligned}\nabla_\theta J(\theta) = &\sum_{i=1}^{N}\sum_{t=1}^{T} Q(\{X^{(i)}, Y_{1:t-1}^{(i)}\}, y_t^{(i)}) \\ &* \nabla_\theta \log G_\theta(y_t^{(i)}|X^{(i)}, Y_{1:t-1}^{(i)}).\end{aligned} \tag{9}$$

According to Eqn. (7),

$$\begin{aligned}&\mathbb{E}_{Y\sim G_\theta(Y|X)}[\sum_{t=1}^{T} Q(\{X, Y_{1:t-1}\}, y_t)] \\ =&\mathbb{E}_{Y\sim G_\theta(Y|X)}[\sum_{t=1}^{T} \mathbb{E}_{Z\sim G_\theta(\cdot|X,Y_{1:t})}[D(X, \{Y_{1:t}, Z\})]] \\ =&\sum_{t=1}^{T} \mathbb{E}_{Y\sim G_\theta(Y|X)} \mathbb{E}_{Z\sim G_\theta(\cdot|X,Y_{1:t})}[D(X, \{Y_{1:t}, Z\})] \\ =&\sum_{t=1}^{T} \mathbb{E}_{Y\sim G_\theta(Y|X)} D(X, Y) = \mathbb{E}_{Y\sim G_\theta(Y|X)}[T \cdot D(X, Y)].\end{aligned} \tag{10}$$

With one-sample estimation of Eqn. (10), we have

$$D(X, Y) = \frac{1}{T}\sum_{t=1}^{T} Q(\{X, Y_{1:t-1}\}, y_t). \tag{11}$$

In this paper, the discriminator is an image-to-sequence model that takes images as the encoder input and Latex sequences as the decoder input, while the output is a sequence of expected future return for each token. The average value of this sequence is the classification of the discriminator. Typically, we train the discriminator using three kinds of image-formula pairs: ground truth sequences with matched images (X, Y), generated sequences with matched images $(X, \hat{Y})$, and ground truth sequences with mismatched ground truth

images $(X, \bar{Y})$. Last, the objective for learning the discriminator can be formulated as follows:

$$L_\phi = \mathbb{E}_{(X,Y)}\left[\log D(X,Y)\right] + \lambda \mathbb{E}_{(X,\bar{Y})}\left[\log(1-D(X,\bar{Y}))\right] + (1-\lambda)\mathbb{E}_{(X,\hat{Y})}\left[\log(1-D(X,\hat{Y}))\right], \quad (12)$$

where λ is a trade-off parameter.

5 EXPERIMENTS

Following [6], we evaluate the proposed method on the IM2LATEX-100K dataset [10] which contains 103,556 formula images of mathematical expressions with their ground-truth Latex sequences.

Dataset and preprocessing. IM2LATEX-100K dataset is extracted from the 2003 KDD cup [15] by parsing Latex sources of papers. The dataset has been split into training (83,883 formulas), validation (9,319 formulas), and test (10,354) sets [10]. The lengths of Latex formulas in IM2LATEX-100K range from 38 to 997 characters, with a mean of 118 and a median of 98.

We build a token vocabulary using formulas in the training set and treat Latex words as tokens (e.g., "\pi" and "\begin{array}") rather than single characters. Those tokens appear less than 10 times are replaced with "UNK" token. Two special tokens, i.e., "START" and "END", are added to the vocabulary to represent the start and the end of the Latex sequences. Our resulting vocabulary size is 499. Following [10], images of large sizes, Latex formulas with more than 150 tokens, or those that cannot be parsed are ignored during training and validation, but included during testing.

Baseline Methods. We compare our method with existing MER systems, including (1) **commercial expression recognition system**: INFTY reader [29] is a commercial mathematical expression recognition system, containing symbol recognition and structural analysis; (2) **classical OCR method**: CRNN [26] is a CTC-based method, using CTC to address the left-to-right ordering assumption; (3) **attention-based methods**: Caption [35], Densenet [33], WYGIWYS [10] and FGFE [6]. These deep learning methods use an attention mechanism to replace the inefficient CTC.

Evaluation Metric. Following [6], we evaluate our method with two kinds of evaluation metrics, i.e., text-based and image-based metrics. The text-based metric includes BLEU and Token Edit Distance (ED_T), which measures the distances between the generated sequences and the ground truth sequences. The image-based metric serves to evaluate the distances between the parsed images of the generated sequences and the ground truth sequences, including the Image Exact Match Accuracy (EM_I) and Image Edit Distance (ED_I).

Implementation Details. We implement the proposed method based on Tensorflow [1]. For a fair comparison, we use the same image feature extractor as [6], and we ignore those images with a size larger than (160, 500) and those Latex formulas with more than 150 tokens during training and validation. For the structure-aware module, the function $f(\cdot)$, $g(\cdot)$ and $h(\cdot)$ are one convolution layer with kernel size 1, and their output channels are 16, 16 and 64. The decoder is a transformer in a *small* configuration [31] with 6 self-attention layers. We pre-train our model for 20 epochs with Adam optimizer using a softmax cross-entropy loss. Batch size, initial learning rate and decay rate are set to 20, 1e-3 and 0.1, respectively. For warming up, we adopt a learning rate of 1e-4 for the first two epochs. For the sequence-level modeling, we train the model for 5 epochs with a learning rate of 5e-5. At test time, we employ the beam search with beam size 5.

Method	BLEU	ED_T	ED_I	EM	EM(w/o space)
CRNN [26]	30.36	-	-	7.60	9.16
INFTY [29]	66.65	-	-	15.60	26.66
Caption [35]	75.01	-	-	53.53	55.72
Densenet [33]	79.21	81.41	76.15	54.54	58.41
WYGIWYS [10]	87.73	87.60	87.90	77.46	79.88
FGFE [6]	87.21	85.96	87.90	77.64	81.71
SASL (ours)	**88.77**	88.66	89.16	**79.37**	**82.59**

Table 1: Comparisons with the state-of-the-arts on IM2LATEX-100.

Method	EM	EM(w/o space)
Densenet [33]	54.90	58.79
WYGIWYS [10]	80.00	82.00
FGFE [6]	80.13	84.30
MER (ours)	81.39	84.47
SAMER (ours)	81.58	85.04
SASL (ours)	**82.23**	**85.54**

Table 2: Comparisons on IM2LATEX-100 (formula length $\leq$ 150).

In MER, the spatial relationship among symbols spans in different directions. To model the spatial relationship, we extend the 1D positional encoding in [31] to two-dimension via concatenating the two positional encodings along the channel dimension. The positional embedding is added to the feature map before the decoder. In our experiment, "MER" denotes the basic model that consists of a convolutional neural network and a Transformer decoder. "SAMER" is the basic model equipped with our proposed structure-aware module. "SASL" is our proposed structure-aware model with sequence-level modeling.

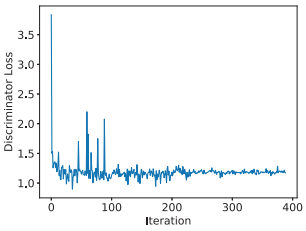
5.1 Comparison with State-of-the-art Methods

We compare our method with state-of-the-arts and report the results in Table 1. From the results, we can draw the following observations: (1) CRNN, a CTC-based model, has the worst performance on both image-based and text-based evaluation metrics. (2) The INFTY achieves a high text-based accuracy but performs poorly on image-based accuracy. (3) All attention-based methods outperform CRNN by a large margin. Overall, compared with existing methods, our method achieves the highest performance on image-based evaluation metric, indicating that our method is able to generate exactly correct Latex sequences. Since our model is trained on those formulas with length less than 150, we further report the results on these data. As shown in Table 2, our model shows superiority over existing methods. We show some prediction results of our model in Figure 3. In Fig. 3, the left column shows the input images, the resulting Latex sequences are shown in the right column. Our model can properly recognize the complex structures, e.g., fraction, matrix, in mathematical expressions. Furthermore, although our proposed method did not have any constraints on generating valid syntax, our model achieves 99.1% valid syntax on test set.

Intense reward fluctuation of discriminator. The fluctuation commonly appears in adversarial learning. To stabilize the

Input Image	Predicted Sequence
$i\sqrt{2}\partial_{-\chi} - g[\phi, \psi] = 0, \partial^2_{-} \bar{A}_{+} - g^2 J^{+} = 0.$	i \sqrt { 2 } \partial _ { - } \chi - g [\phi , \psi] = 0 , \quad \partial _ { - } ^ { 2 } \bar { A } _ { + } - g ^ { 2 } J ^ { + } = 0 .
$R(e_1) = \epsilon^{-J_{67}+J_{89}}, \quad R(e_2) = \epsilon^{J_{45}-J_{89}}.$	R (e _ { 1 }) = \epsilon ^ { - J _ { 6 7 } + J _ { 8 9 } } , \quad R (e _ { 2 }) = \epsilon ^ { J _ { 4 5 } - J _ { 8 9 } } .
$\rho^0 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$ and $\rho^1 = \begin{pmatrix} 0 & i \\ i & 0 \end{pmatrix}.$	\rho ^ { 0 } = \left(\begin{array} { c c } { 0 } & { - i } \\ { i } & { 0 } \end{array} \right) \; \; \mathrm { a n d } \; \; \rho ^ { 1 } = \left(\begin{array} { c c } { 0 } & { i } \\ { i } & { 0 } \end{array} \right) .
$Q = c\sum_i f_i' p^i + \sum_k c_k p^k f_k + ifinite\ more.$	Q = c \sum _ { i } f _ { i } ^ { \prime } p ^ { i } + \sum _ { k } c _ { k } p ^ { k } f _ { k } + i n f i n i t e \; m o r e .
$S \rightarrow S + \frac{i\left[\left(\tilde{U}^T\right)^{-1}\right]_{1^\Sigma}\left(H_\Sigma^{(1)} + C_{\Sigma\Delta}\hat{X}^\Delta\right)}{\tilde{U}^0_\Lambda \hat{X}^\Lambda}.$	S \rightarrow S + \frac { i [(\hat { U } ^ { T }) ^ { - 1 }] _ { 1 } ^ { \ \Sigma } (H _ { \Sigma } ^ { (1) } + C _ { \Sigma \Delta } \hat { X } ^ { \Delta }) } { \hat { U } _ { \Lambda } ^ { 0 } \hat { X } ^ { \Lambda } } .
$L'_g(v(h)) = v(L_g h) = v(gh), \forall g, h \in G,$	L _ { g } ^ { ' } \Bigl (v (h) \Bigr) = v (L _ { g } h) = v (g h) \, , \forall g , h \in G ,
$E_{12}\ \Phi = 2\sqrt{\left(m + \frac{1}{2}br\right)^2 + p_r^2 + \frac{l(l+1)}{r^2}}\ \Phi,$	E _ { 1 2 } \sim \sim \Phi = 2 \sqrt { (m + \frac { 1 } { 2 } b r) ^ { 2 } + p _ { r } ^ { 2 } + \frac { \ell (\ell + 1) } { r ^ { 2 } } } \sim \sim \Phi ,

Figure 3: Examples of Prediction Results.



(a)

$\alpha_1^r\gamma_1 + \ldots + \alpha_N^r\gamma_N = 0$

$\alpha_1^r\gamma_1 + \ldots + \alpha_N^r\gamma_N = 0$

$\alpha_1^r\gamma_1 + \ldots + \alpha_N^r\gamma_N = 0$

$\alpha_1^r\gamma_1 + \ldots + \alpha_N^r\gamma_N = 0$

$\alpha_1^r\gamma_1 + \ldots + \alpha_N^r\gamma_N = 0$

(b)

Figure 4: (a) Learning curve of discriminator. (b) Qualitative examples of the attention mechanism.

training process, we pretrain the discriminator. We also feed ground truth sequence to the generator, and set reward to 1 for model update. With this strategy, even if the generator gets lost, it knows what sequences are good and how to push itself to generate these good sequences. Besides, we show the training loss of discriminator in Figure 4(a). As shown in the Figure, the discriminator finally converges. Note that the feature extractor of discriminator was initialized with the weights of the pretrained generator, thus the loss of discriminator decreases rapidly at the beginning.

Attention Visualization. Our method uses the attention mechanism to sequentially focus on informative characters on the images, which implicitly assumes a left-to-right order. To better understand the generation process, we visualize the encoder-decoder attention maps to show the translation process of our model in Figure 4(b). An image region with a higher attention weight is masked with red square, which indicates where the model concentrates on. Each line in Fig.4 is related to a character, e.g., the first line is "\alpha" and the second line is "1". As shown in Figure 4(b), our model can sequentially attend the characters in left-to-right order.

5.2 Ablation Studies

To verify the effectiveness of each component in our method, we conduct ablation studies by removing different components from our model. We show experimental results in Table 3. The simplest model, denoted as "MER w/o pos", is the combination of a basic convolutional neural network and a Transformer decoder, which achieves 75.10% on Image Exact Match (EM). Adding the positional encoding to the image feature improves the performance of the EM to 78.87%, indicating that the positional information provides useful position cues to facilitate sequence decoding. Furthermore, with the help of the structure relationship modeling module, our method achieves 79.07% in EM. With the sequence-level modeling, the EM score of our method is further raised to 79.37%.

Method	BLEU	ED_T	ED_I	EM	EM(w/o space)
MER w/o pos	83.92	86.91	89.37	75.10	78.52
MER	87.99	88.81	89.81	78.87	81.86
SAMER	88.06	**89.13**	88.92	79.07	82.45
SASL	**88.77**	88.66	**89.16**	**79.37**	**82.59**

Table 3: Ablation studies on different components.

λ	BLEU	ED_T	ED_I	EM	EM(w/o space)
0.1	88.26	89.80	91.40	78.05	81.36
0.3	88.77	88.66	89.16	79.37	82.59
0.5	88.01	89.96	88.37	78.08	81.25

Table 4: Performance comparisons with different λ.

5.3 Parameter Sensitivity Analysis

In this section, we investigate how the proposed method performs with the changes of the hyper-parameter. We set the ratio of different types of mismatch pairs $\lambda = \{0.1, 0.3, 0.5\}$. The experimental result is shown in Table 4. Since we focus on the image-based evaluation metric, we choose $\lambda = 0.3$ in our experiments.

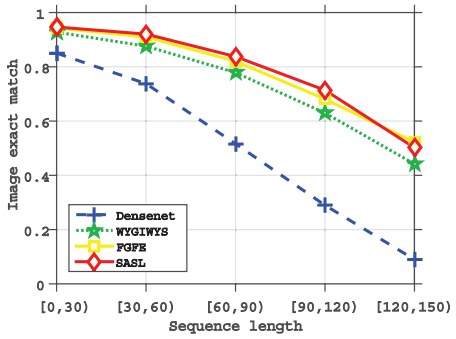


Figure 5: The impact of formula length on image exact match.

PD: Eq.(1) $\simeq ABC + \frac{i}{2}\left[\frac{\partial A}{\partial P^{\mu}}\frac{\partial B}{\partial X_{\mu}}C - A\frac{\partial B}{\partial X_{\mu}}\frac{\partial C}{\partial P^{\mu}} - \frac{\partial A}{\partial X_{\mu}}\frac{\partial BC}{\partial P^{\mu}} + \frac{\partial AB}{\partial P^{\mu}}\right]$

GT: Eq.(1) $\simeq ABC + \frac{i}{2}\left[\frac{\partial A}{\partial P^{\mu}}\frac{\partial B}{\partial X_{\mu}}C - A\frac{\partial B}{\partial X_{\mu}}\frac{\partial C}{\partial P^{\mu}} - \frac{\partial A}{\partial X_{\mu}}\frac{\partial BC}{\partial P^{\mu}} + \frac{\partial AB}{\partial P^{\mu}}\boxed{\frac{\partial C}{\partial X_{\mu}}}\right]$

PD: $\Delta^{(N,0)}(s) = -\sum_{n>0,\vec{n}^2<N}[J(z_n) - 2 + 2J(y_n) + \frac{J^2(y_n)}{2(1-y_n)} - J(\tilde{z_n})]$

GT: $\Delta^{(N,0)}(s) = -\sum_{n>0,\vec{n}^2<N}[J(z_n) - 2 + 2J(y_n) + \frac{J^2(y_n)}{2(1-y_n)} - J(\tilde{z}_n)\boxed{-2J(\tilde{y}_n)}]$

Figure 6: Samples of long length formulas generation results. "GT" denotes ground truth and "PD" denotes prediction by our model.

5.4 Failure Case Analysis

In this section, we analyze some failure cases of our model. We find that our model fails on those formulas with long-length and images with multi-line expressions.

Long Length Expression Recognition. To show our model's ability to generate Latex sequences with different lengths, we group the test set according to their sequence length and evaluate the image exact match score. To be consistent with the training data, we only show the results on those formulas with length less than 150. As shown in Figure 5, all models experience a performance drop on longer expressions. We show some examples in Figure 6. Although our model failed to make a correct prediction at the end of the formula, it correctly recognized the fraction, super-script and sub-script. To address this issue, one possible solution is to clip the formula image into some images with short width, and then generate Latex sequences respectively.

Multi-line Expression Recognition. Our model is trained on those images with a height less than 160 and a width less than 500. However, the test set includes many large images with 160 $\leq$ height, width $\leq$ 800, which often contain multiple lines of expressions. We find it is hard for our model to recognize these multi-line expressions. For example, our method fails to recognize the multi-line structure in Figure 7 (b). Formulas in the training set contain at most two lines, which may be the reason that the model tends to capture the first two lines for Fig.7. To tackle this problem, we propose to decompose the multi-line expression recognition into multiple single-line expression recognition tasks. Specifically, we first horizontally project the image [27]. Then, we find the upper and lower limits of each line, and perform line cutting. Last, we perform expression recognition for each clipped formula image.

(a) Ground truth image

$$\begin{array}{ll} F_{0A}+\lambda_0 F_{A3}=0, & \lambda_0 F_{12}+\lambda_1 F_{45}+\lambda_2 F_{67}=0, \\ F_{14}+\lambda_0\lambda_1 F_{52}=0, & F_{15}+\lambda_0\lambda_1 F_{24}=0, \\ F_{16}+\lambda_0\lambda_2 F_{72}=0, & F_{17}+\lambda_0\lambda_2 F_{26}=0, \\ F_{18}+\lambda_0\lambda_1\lambda_2 F_{92}=0, & F_{19}+\lambda_0\lambda_1\lambda_2 F_{28}=0, \\ F_{46}+\lambda_1\lambda_2 F_{75}=0, & F_{47}+\lambda_1\lambda_2 F_{56}=0, \\ F_{48}+\lambda_2 F_{95}=0, & F_{49}+\lambda_2 F_{58}=0, \\ F_{68}+\lambda_1 F_{97}=0, & F_{69}+\lambda_1 F_{78}=0. \end{array}$$

(b) Predicted image

$$F_{0A}+\lambda_0 F_{A3}=0, \quad F_{15}+\lambda_0\lambda_1 F_{24}=0$$

(c) Predicted image (after clipping)

$$\begin{array}{ll} F_{0A}+\lambda_0 F_{A3}=0, & \lambda_0 F_{12}+\lambda_1 F_{45}+\lambda_2 F_{67}=0, \\ F_{14}+\lambda_0\lambda_1 F_{52}=0, & F_{15}+\lambda_0\lambda_1 F_{24}=0, \\ F_{16}+\lambda_0\lambda_2 F_{72}=0, & F_{17}+\lambda_0\lambda_2 F_{26}=0, \\ F_{18}+\lambda_0\lambda_1\lambda_2 F_{92}=0, & F_{19}+\lambda_0\lambda_1\lambda_2 F_{28}=0, \\ F_{46}+\lambda_1\lambda_2 F_{75}=0, & F_{47}+\lambda_1\lambda_2 F_{56}=0, \\ F_{48}+\lambda_2 F_{95}=0, & F_{49}+\lambda_2 F_{58}=0, \\ F_{68}+\lambda_1 F_{97}=0, & F_{69}+\lambda_1 F_{78}=0. \end{array}$$

Figure 7: An example of multi-line expression recognition.

As shown in Figure 7 (c), our method obtains a large performance improvement, although the typesetting of output expressions may not align with the ground truth.

Although our model performs poorly on multi-line expressions, our proposed method improves the EM score from 80.92% to 81.98% on those multi-line expressions.

6 CONCLUSION

In this paper, we have proposed a structure-aware mathematical expression recognition method with sequence-level modeling (SASL). Our method contains a structure-aware module to deal with the complex structural analysis. The structure-aware module serves to model the relationship among image regions by measuring the similarity between different image regions. Our model is trained at a sequence-level. Specially, we model the Latex sequences generation process as a Markov Decision Process and solve it using reinforcement learning. The reward signal is provided by a trained discriminator, which evaluates how well the generated sequences match the input image. Specifically, we propose a stepwise evaluation for Latex sequences to stabilize policy learning. Extensive experimental results on the IM2LATEX-100K dataset demonstrate that our method outperforms the state-of-the-arts. Note that our method is also able to be transfered to handwritten MER (HMER), since the only difference between printed MER and HMER is the image feature extraction, which is left as our future work.

Acknowledgements. This work was partially supported by the Ministry of Science and Technology Foundation Project (2020AAA0106901), National Natural Science Foundation of China (NSFC) 62072190, Program for Guangdong Introducing Innovative and Enterpreneurial Teams 2017ZT07X183, Key-Area Research and Development Program of Guangdong Province 2018B010108002, Fundamental Research Funds for the Central Universities D2191240.

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *USENIX Symposium on Operating Systems Design and Implementation*. 265–283.

[2] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. SPICE: Semantic Propositional Image Caption Evaluation. In *European Conference on Computer Vision*. 382–398.

[3] Ahmad-Montaser Awal, Harold Mouchere, and Christian Viard-Gaudin. 2009. Towards handwritten mathematical expression recognition. In *International Conference on Document Analysis and Recognition*. 1046–1050.

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *ArXiv* (2014).

[5] Abdelwaheb Belaid and Jean-Paul Haton. 1984. A syntactic approach for handwritten mathematical formula recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (1984), 105–111.

[6] Sidney Bender, Monica Haurilet, Alina Roitberg, and Rainer Stiefelhagen. 2019. Learning Fine-Grained Image Representations for Mathematical Expression Recognition. In *International Conference on Document Analysis and Recognition Workshops*. 56–61.

[7] Kam-Fai Chan and Dit-Yan Yeung. 2000. Mathematical expression recognition: a survey. *International Journal on Document Analysis and Recognition* (2000), 3–15.

[8] Chen Chen, Shuai Mu, Wanpeng Xiao, Zexiong Ye, Liesi Wu, and Qi Ju. 2019. Improving image captioning with conditional generative adversarial nets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 8142–8150.

[9] Chen Chen, Shuai Mu, Wanpeng Xiao, Zexiong Ye, Liesi Wu, Fuming Ma, and Q. Ju. 2019. Improving Image Captioning with Conditional Generative Adversarial Nets. In *AAAI Conference on Artificial Intelligence*. 8142–8150.

[10] Deng et.al. 2016. Image-to-markup generation with coarse-to-fine attention. *ArXiv* (2016).

[11] Zhang et.al. [n.d.]. Multi-scale attention with dense encoder for hand written mathematical expression recognition. In *2018ICPR*.

[12] Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C Platt, et al. 2015. From captions to visual concepts and back. In *IEEE Conference on Computer Vision and Pattern Recognition*. 1473–1482.

[13] Chuang Gan, Zhe Gan, Xiaodong He, Jianfeng Gao, and Li Deng. 2017. Stylenet: Generating attractive visual captions with styles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3137–3146.

[14] Zhe Gan, Chuang Gan, Xiaodong He, Yunchen Pu, Kenneth Tran, Jianfeng Gao, Lawrence Carin, and Li Deng. 2017. Semantic compositional networks for visual captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5630–5639.

[15] Johannes Gehrke, Paul Ginsparg, and Jon Kleinberg. 2003. Overview of the 2003 KDD Cup. *Acm Sigkdd Explorations Newsletter* 5, 2 (2003), 149–151.

[16] Zhiting Hu, Zichao Yang, Xiaodan Liang, R. Salakhutdinov, and E. Xing. 2017. Toward Controlled Generation of Text. In *International Conference on Machine Learning*. 1587–1596.

[17] Giorgos Kordopatis-Zilos, Symeon Papadopoulos, Ioannis Patras, and Ioannis Kompatsiaris. 2019. ViSiL: Fine-grained spatio-temporal video similarity learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 6351–6360.

[18] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. 2019. Similarity of neural network representations revisited. In *International Conference on Machine Learning*. PMLR, 3519–3529.

[19] Stéphane Lavirotte and Loic Pottier. 1998. Mathematical formula recognition using graph grammar. In *Document Recognition V*. 44–52.

[20] Anh Duc Le and Masaki Nakagawa. 2017. Training an End-to-End System for Handwritten Mathematical Expression Recognition by Generated Patterns. *International Conference on Document Analysis and Recognition* (2017), 1056–1061.

[21] Chang Liu, Fuchun Sun, Changhu Wang, Feng Wang, and Alan Yuille. 2017. MAT: A multimodal attentive translator for image captioning. In *International Joint Conference on Artificial Intelligence*. 4033–4039.

[22] Shubo Ma and Yahong Han. 2016. Describing images by feeding LSTM with structural words. In *IEEE International Conference on Multimedia and Expo*. 1–6.

[23] Erik G Miller and Paul A Viola. 1998. Ambiguity and constraint in mathematical expression recognition. In *AAAI/IAAI*. 784–791.

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

[25] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7008–7024.

[26] Baoguang Shi, Xiang Bai, and Cong Yao. 2016. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2016), 2298–2304.

[27] Archana A Shinde and DG Chougule. 2012. Text pre-processing and text segmentation for OCR. *International Journal of Computer Science Engineering and Technology* 2, 1 (2012), 810–812.

[28] Richard Socher, Andrej Karpathy, Quoc V Le, Christopher D Manning, and Andrew Y Ng. 2014. Grounded compositional semantics for finding and describing images with sentences. *Transactions of Association for Computational Linguistics* (2014), 207–218.

[29] Masakazu Suzuki, Fumikazu Tamari, Ryoji Fukuda, Seiichi Uchida, and Toshihiro Kanahori. 2003. INFTY: an integrated OCR system for mathematical documents. In *ACM Symposium on Document Engineering*. 95–104.

[30] Zhi Tian, Weilin Huang, Tong He, Pan He, and Yu Qiao. 2016. Detecting text in natural image with connectionist text proposal network. In *European Conference on Computer Vision*. 56–72.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. 5998–6008.

[32] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. CIDEr: Consensus-based image description evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition*. 4566–4575.

[33] Jian Wang, Yunchuan Sun, and Shenling Wang. 2019. Image to latex with densenet encoder and joint attention. *Procedia Computer Science* (2019), 374–380.

[34] Guanghui Xu, Shuaicheng Niu, Mingkui Tan, Yucheng Luo, Qing Du, and Qi Wu. 2021. Towards Accurate Text-based Image Captioning with Content Diversity Exploration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12637–12646.

[35] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*. 2048–2057.

[36] Ke Xu, Yifan Zhang, Deheng Ye, Peilin Zhao, and Mingkui Tan. 2020. Relation-Aware Transformer for Portfolio Policy Learning. In *International Joint Conference on Artificial Intelligence*. 4647–4653.

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

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

[39] Ting Zhang. 2017. *New Architectures for Handwritten Mathematical Expressions Recognition.* Ph.D. Dissertation.

[40] Wei Zhang, Zhiqiang Bai, and Yuesheng Zhu. 2019. An improved approach based on CNN-RNNs for mathematical expression recognition. In *International Conference on Multimedia Systems and Signal Processing*. 57–61.

[41] Yifan Zhang, Peilin Zhao, Bin Li, et al. 2020. Cost-sensitive portfolio selection via deep reinforcement learning. *IEEE Transactions on Knowledge and Data Engineering* (2020).