Pattern Recognition 140 (2023) 109538

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Joint optimization for attention-based generation and recognition of chinese characters using tree position embedding[☆]

Mobai Xue, Jun Du[*], Bin Wang, Bo Ren, Yu Hu

National Engineering Research Center of Speech and Language Information Processing (NERC-SLIP), University of Science and Technology of China, No. 96, JinZhai Road, Hefei, Anhui, PR China

ARTICLE INFO

Article history:
Received 26 April 2022
Revised 14 March 2023
Accepted 19 March 2023
Available online 21 March 2023

Keywords:
Chinese character generation and recognition
Radical analysis
Joint optimization
Tree position embedding

ABSTRACT

Despite the growing interest in Chinese character generation, creating a nonexistent character remains an open challenge. Radical-based Chinese character generation is still a novel task while radical-based Chinese character recognition is more technologically advanced. To fully utilize the knowledge of recognition task, we first propose an attention-based generator. The generator chooses the most relevant radical to generate each zone with an attention mechanism. Then, we present a joint optimization approach to training generation-recognition models, which can help the generator and recognizer learn from each other effectively. The joint optimization is implemented via contrastive learning and dual learning. Considering the symmetry of the generation and recognition, contrastive learning aims to strengthen the performance of the encoder of recognizer and the decoder of generator. Since the generation and recognition tasks can form a closed loop, dual learning feeds the output from one to another as input. Based on the feedback signals generated during the two tasks, we can iteratively update the two models until convergence. Finally, as our model ignores the order information of a sequence, we exploit position embedding to extend the image representation ability and propose tree position embedding to represent the positional information for tree structure captions of Chinese characters. The experimental results in printed and nature scenes show that the proposed method improves the quality of the generating images and increases the recognition accuracy for Chinese characters.

1. Introduction

In our daily lives, automatic recognition of Chinese text is frequently employed. However, the recognition of Chinese characters or texts is among the most challenging topics in pattern recognition, due to the numerous character categories, complex internal structures and confusion among similar characters. Additionally, recognition of rarely used Chinese characters is typically a few-shot learning problem since samples of such character categories are difficult to collect. Moreover, recognition of some novel Chinese characters is a zero-shot learning problem, as these characters are newly created and have never been seen before.

In this case, a radical-based model is suggested that fully exploits the highly hierarchical structure of Chinese characters. Only roughly 500 radicals and 10 spatial structures are sufficient to describe the more than 20,000 Chinese characters because all Chinese characters are made up of radicals. Ten spatial structures are shown in Fig. 1 while a Chinese character's caption sequence, tree view, and spatial structure diagram are shown in Fig. 2. The radical-based method aims to decompose Chinese characters into radicals and describe their spatial structures as captions for recognition.

However, the radical-based recognition method still needs to be improved in the complex Chinese scene. In addition to improving the recognition model itself, data augmentation and dual learning have also been used recently. Geirhos et al. [1] used the Chinese characters with new styles generated by style transfer as data augmentation to expand the training set. Zhu et al. [2] employed a style transfer model as the generator and formed a dual system with the recognizer. However, style transfer is not the reverse task of recognition, which plays a limited role in dual learning. What's more, both the above methods are not helpful for the recognition of novel Chinese characters. On this basis, we propose a radical-based generator that can create Chinese characters from texts, since a character-based model cannot identify a novel character. And then, we design a joint optimization for generation-

Abbreviations: RTN, Radical and Transformer based Network; TPE, Tree Position Embedding.

[☆] Fully documented templates are available in the elsarticle package on CTAN.

[*] Corresponding author.

E-mail address: jundu@ustc.edu.cn (J. Du).

https://doi.org/10.1016/j.patcog.2023.109538
0031-3203/© 2023 Elsevier Ltd. All rights reserved.

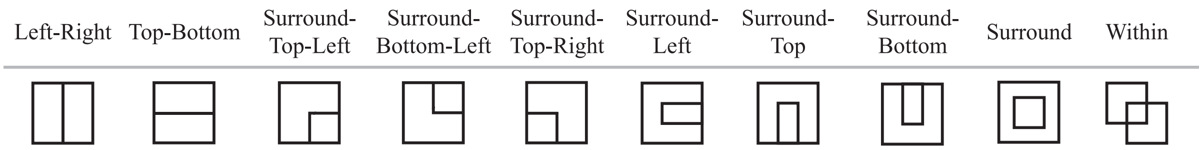
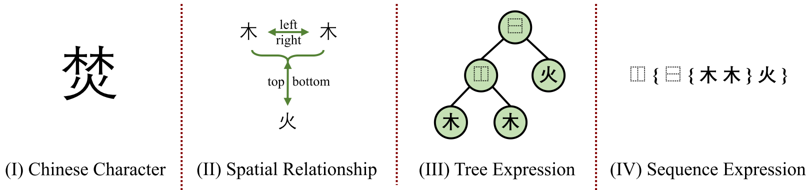


Fig. 1. Graphical representation of ten common spatial structures between the radicals of Chinese characters.



(I) Chinese Character (II) Spatial Relationship (III) Tree Expression (IV) Sequence Expression

Fig. 2. In the radical analysis, each Chinese character is equal to a tree, and each tree is stored as a sequence according to the depth-first traversal order.

recognition system and employ the generated characters as data augmentation in experiments.

Radical-based Chinese character generation is a novel task due to the diversified character radicals, variable font styles and complicated foreground and background. It is challenging for the model to generate high-quality characters in complex scenes. Recently, a radical combination network (RCN) [3] is proposed, which can generate a Chinese character with caption input. The key idea is to integrate the information of the radicals and character attributes as the representation by a recursive network. However, since the recursive model operates the radicals one by one, the preceding tokens are covered by the succeeding ones in a sequence. The information of leaf nodes in bottom layers is largely ignored and the outputs are most affected by the information of root nodes in top layers, which implies the weak capacity of dealing with long-distance dependency. Moreover, the generalization of RCN and the quality of the generated results on hand-writing scene are not always satisfactory.

To address this issue, we first propose a radical-and-transformer generation network (RTN-G) for Chinese character generation. The information in bottom layer and top layers are simultaneously calculated instead of one by one, which performs better on processing long-distance information than recursive network [4]. The representations of radicals are fed to a Transformer network to build contextualized representations and the model is trained to draw Chinese characters according to their attributes. Compared with RCN, RTN-G can produce higher-quality images of Chinese characters with more precise model building. Second, to make a full use of the prior knowledge of Chinese character recognition, we employ an attention-based recognizer (RTN-R) to form a set of symmetric networks with the generator. The recognition model consists of a fully convolutional network as the encoder and a transformer decoder. Our model framework is shown in Fig. 4. The radical-and-transformer based recognition network performs a recognition task in a symmetrical way while the generator (RTN-G) and recognizer (RTN-R) can form a closed loop. Then we jointly optimize recognition and generation models by leveraging the strong complementarity between two tasks. Considering the symmetry of the architecture, we adopt contrastive learning and dual learning to assist the generator and recognizer in learning from each other:

- **Contrastive Learning**: Considering the symmetry of recognizer and generator, the representations in symmetrical position have the same physical meaning. Given a representation, the model improves the ability of modeling by identifying the symmetrical one in a set of candidate representations.
- **Dual Learning**: The generator and recognizer can be an evaluator of each other [5]. The generator and recognizer can be optimized with the help of each other and they can well collaborate to improve the performance for both.

In addition, the generator creates Chinese characters from texts, which means the non-existed Chinese characters can be produced. The non-existed Chinese characters contains a large number of novel radical combinations and can effectively expand the data set. More combinations might help the recognition models distinguish radicals. In the experiments, non-existed characters were employed and the model correctly recognized easily confused Chinese characters.

Since our model contains no recurrence, we employ position embedding to represent the positional information of radicals in images. Considering the significant difference between sequence and tree structure, we propose a tree position embedding (TPE) for tree structure captions of Chinese characters to achieve a better structure modeling for both Chinese character generation and recognition tasks.

The main contributions of this study are summarized as follows:

- We propose an attention-based Chinese character generation network. In particular, the network well generates the alignment strongly corresponding to human intuition. Experimental results show that our method performs better than the recursive model.
- A joint optimization method is presented for Chinese generation and recognition tasks by using contrastive learning and dual learning. Experimental results demonstrate that it further improves the quality of the generated Chinese characters and increases the recognition accuracy of Chinese characters in both printed and natural scenes.
- We propose a tree position embedding (TPE), which performs better than 1D position embedding (PE). We empirically prove the superiority of the TPE over the traditional PE for representing the tree structure information.

2. Related works

2.1. Radical-based recognition

The recognition of Chinese characters is an intricate problem due to a large number of existing character categories. However, most recognition methods are character-based and can only recognize about 4000 commonly used characters. Compared with the character-based method treating characters as independent ones, the radical-based method can summarize and extract the similarity among Chinese characters. As shown in Fig. 3, radical-based method can recognize out-of-dictionary characters but character-

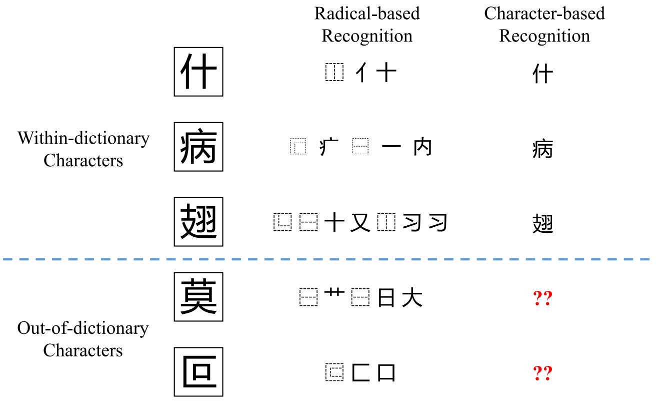

Radical-based Recognition
Character-based Recognition
Within-dictionary Characters
Out-of-dictionary Characters
什
病
翊
莫
叵
??
??

Fig. 3. The comparison between the radical-based and character-based recognition methods. Radical-based methods can recognize out-of-dictionary characters but the character-based cannot.

based cannot. Therefore, radical-based recognition is more valuable and performs better on Chinese character recognition. In the past few decades, lots of efforts have been made for radical-based Chinese character recognition. Wang and Fan [6] employed a hierarchical radical matching method to recognize a Chinese character. Shi et al. [7] used a HMM-based model and [8] used an BLSTM-CRF neural architecture for Chinese character recognition. RTN-R [9] proposed a Transformer-based encoder-decoder model [10] for radical-based Chinese character recognition. TAN [11] employs a tree decoder, which computes a relation node and two radical nodes at each step. Compared with string decoders, the tree decoder better understands tree structures but costs more. In our study, we employ the framework of RTN-R as the recognizer.

2.2. Radical-based generation

Chinese character generation is a challenging task. The most of existing works generate Chinese characters in character level with complex background and font styles. Lin et al. [12] generated nearly real Chinese characters using a GAN model in character level. However, in addition to the above-mentioned, radical level generator can create Chinese characters with novel classes, which is an artistic work and can provide variable combination of radicals as data augmentation for radical-based recognition. Huang et al. [13] proposed a novel radical decomposition and rendering based GAN to transfer the font styles. Zhang et al. [14] used writing trajectories to generate handwritten Chinese characters. Recently, RCN [3] proposed a radical-based generator, which can generate Chinese characters by using radicals. In our study, we propose an attention-based encoder-decoder network (RTN-G) as the generator and achieve radical-based generation for Chinese characters.

2.3. Joint optimization

Joint optimization is commonly used for multi-tasks within the same framework [15]. In this paper, we propose a cross-model joint optimization for generation and recognition, which consists of contrastive learning and dual learning [16]. Contrastive learning has recently achieved success [17]. Many such approaches have relied on heuristics to design pretext tasks [18]. Discriminative approaches based on contrastive learning in the latent space have recently shown great promise, achieving state-of-the-art results [19]. Aberdam et al. [20] proposes a contrastive learning approach to sequential prediction tasks such as text recognition. Baevski et al. [21] masks the speech input in the latent space and solves a contrastive task defined over quantization of the latent representations which are jointly learned. In our study, we aim to make the model identify the true representations of characters in a set of candidate representations and also make the attention matrices close to their archor. Dual learning is a training strategy applied to dual tasks and [22] achieves comparable accuracy to neural machine translation trained from the full bilingual data, by learning from monolingual data. Zhu et al. [2] applied dual learning on recognition and generation tasks. In this study, our model improves with this training strategy.

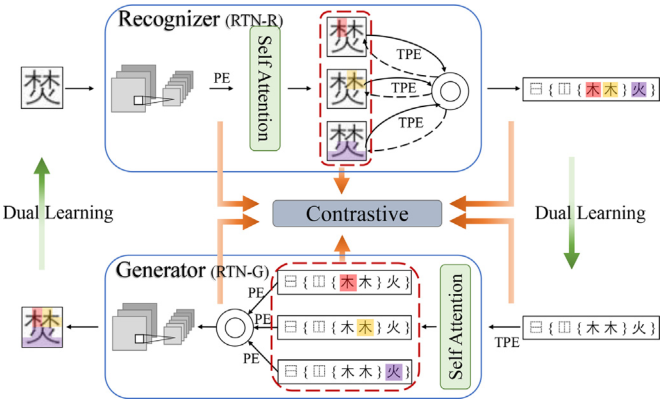


Fig. 4. Illustration of our framework which contains a recognizer and a generator. The two networks perform a recognition task and a generation task respectively. Through feeding the output to the other network, the recognizer and generator can jointly learn mapping relationships between radicals and images. Besides, the model employs contrastive learning for reinforcement of joint learning further. Position embedding (PE) and tree position embedding (TPE) are employed here. The red dotted boxes correspond to the attention maps, the red lines correspond to the process of contrastive learning and the green lines correspond to the process of dual learning.

3. Methodology

In this paper, a joint model based on radical and attention is presented as shown in Fig. 4, which aims to improve the quality of generated characters and the accuracy rate of recognition. Firstly, we propose a radical-based generator to realize the transfer from text to image. Then, we employ a radical-based recognizer which has proven to be highly effective. Finally, we introduce joint optimization including contrastive learning and dual learning. Details of the radical and transformer based generator (RTN-G, radical and transformer based generation network) are described in Section 3.1. Details of the radical and transformer based recognizer (RTN-R, radical and transformer based recognition network) are described in Section 3.2. Section 3.3 illustrates the joint optimization and the loss function.

3.1. Radical-and-transformer-based generation network

On the basis of previous work RCN [3], we propose an attention-based generator, an encoder-decoder network, as shown in the bottom half of Fig. 4. Transformer blocks [4] with L layers are used for the encoder and decoder, and configurations are outlined in Table 1. The deconvolution algorithm is employed to generate the final output images.

Firstly, we map N symbols of source captions $\boldsymbol{S}_C$ to a sequence of radical representations $\boldsymbol{X}$ as encoder input. Each token is finally represented by the summation of the word embedding $\boldsymbol{x}_i$ and the tree position embedding TPE(i).

$$\boldsymbol{X} = (\boldsymbol{x}_1 + \mathrm{TPE}(1), \ldots, \boldsymbol{x}_N + \mathrm{TPE}(N)) \tag{1}$$

where $\boldsymbol{x}_i, \mathrm{TPE}(i) \in \mathbb{R}^D$ and D is the model dimension. We map symbols of character attributes to background feature $\boldsymbol{B}$ as decoder input, which represents writing style, background, color, angle, etc.

Table 1
Encoder and decoder configurations of generator and recognizer. The layers are shown in bold.

Generator		Recognizer	
Encoder Layer: 12	Decoder Layer: 12	Encoder Layer: 6	Decoder Layer: 6
Input			
Word Embeddings $(N \times D)$	Background Feature $(\frac{HW}{T^2} \times D)$	Image Feature $(\frac{HW}{T^2} \times D)$	Masked Word Embedding $(N \times N \times D)$
Multi-Head Attention			
Head: 4	Head: 4	Head: 8	Head: 8
Add & **Norm** **Feed Forward**			
$\boldsymbol{W}_1 \in \mathbb{R}^{D\times D}$ $\boldsymbol{W}_2 \in \mathbb{R}^{D\times D}$	$\boldsymbol{W}_1 \in \mathbb{R}^{D\times 2D}$ $\boldsymbol{W}_2 \in \mathbb{R}^{2D\times D}$	$\boldsymbol{W}_1 \in \mathbb{R}^{D\times D}$ $\boldsymbol{W}_2 \in \mathbb{R}^{D\times D}$	$\boldsymbol{W}_1 \in \mathbb{R}^{D\times 2D}$ $\boldsymbol{W}_2 \in \mathbb{R}^{2D\times D}$
Add & **Norm**			

Each relative position corresponds to a position embedding $\mathrm{PE}(i)$ [23].

$$\boldsymbol{B} = (\boldsymbol{b}_1 + \mathrm{PE}(1), \ldots, \boldsymbol{b}_K + \mathrm{PE}(K)) \tag{2}$$

where $\boldsymbol{b}_i, \mathrm{PE}(i) \in \mathbb{R}^D$, and $K = \frac{HW}{T^2}$ denotes the length of $\boldsymbol{B}$. H and W denote the shape of output images, and T denotes the multiple of sampling. Then, $\boldsymbol{X}$ is fed to the encoder to build context vector $\boldsymbol{C}$. Given $\boldsymbol{C}$ and $\boldsymbol{B}$, the decoder generates character representations $\boldsymbol{Z}$ with fixed-shape. Finally, $\boldsymbol{Z}$ is up-sampled to generate RGB images $\boldsymbol{F}$, where $\boldsymbol{F} \in \mathbb{R}^{H\times W\times 3}$.

3.1.1. Encoder

The encoder is composed of a stack of L identical layers. Each layer has a multi-head self-attention mechanism with M heads [24] and a position-wise feed-forward network.

Multi-head self-attention mechanism is a proved attention mechanism, where self-attention can be described as:

$$\mathrm{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \mathrm{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{D}}\right)\boldsymbol{V} \tag{3}$$

where $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V} \in \mathbb{R}^{N\times D}$ are mapped by encoder input. We use the dot product on $\boldsymbol{Q}$ and $\boldsymbol{V}$ to obtain the correlation matrix as weight and employ a weighted summation on $\boldsymbol{V}$. The multi-head attention can be expressed in the same notation as:

$$\begin{gathered}\mathrm{MultiHead}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \mathrm{Concat}(\mathrm{h}_1, \ldots, \mathrm{h}_M)\boldsymbol{W}_\mathrm{O} \\ \mathrm{h}_i = \mathrm{Attention}(\boldsymbol{Q}\boldsymbol{W}_i^Q, \boldsymbol{K}\boldsymbol{W}_i^K, \boldsymbol{V}\boldsymbol{W}_i^V)\end{gathered} \tag{4}$$

where h_i represents the i^{th} head, $\boldsymbol{W}_i \in \mathbb{R}^{D\times D/M}$ are the projection matrices for the i^{th} head, and $\boldsymbol{W}_\mathrm{O} \in \mathbb{R}^{D\times D}$.

The feed-forward network (FFN) contains two fully connected layers and an active layer. The parameters of multi-head self-attention and FFN are illustrated in Table 1.

$$\mathrm{FFN}(\boldsymbol{x}) = \max(0, \boldsymbol{x}\boldsymbol{W}_1 + \boldsymbol{b}_1)\boldsymbol{W}_2 + \boldsymbol{b}_2 \tag{5}$$

3.1.2. Decoder

The decoder is composed of a stack of L identical layers. Each layer has a multi-head attention mechanism with M heads [25] and a feed-forward network, which is similar to the encoder.

We map encoder output to key $\boldsymbol{K}$ and value $\boldsymbol{V}$ and map decoder input to query $\boldsymbol{Q}$. Regarding the decoder input as a blank canvas, we draw characters according to the relationship between radicals and image zones, which is built by attention. We define the multiplication of attention matrices in different layers as textual attention $\boldsymbol{A}_\mathrm{g}$, which will be used in joint optimization.

$$\boldsymbol{A}_\mathrm{g} = \prod_{j=1}^{L}\sum_{i=1}^{M}\mathrm{softmax}\left(\frac{\boldsymbol{Q}_i^j \boldsymbol{K}_i^{j\top}}{\sqrt{D}}\right) \tag{6}$$

where i denotes the i^{th} head and j denotes the j^{th} layer in decoder. $\boldsymbol{A}_\mathrm{g}$ performs as the weighting coefficients so that it can choose the most relevant radical from the whole input sequence for calculating the context vector.

Given a source caption $\boldsymbol{S}_\mathrm{C}$, background feature $\boldsymbol{B}$ and a target image $\boldsymbol{S}_\mathrm{I}$, the generator can be defined as the following conditional probability:

$$\mathrm{P}(\boldsymbol{S}_\mathrm{I}|\boldsymbol{S}_\mathrm{C}, \boldsymbol{B}) = \prod_{i=1}^{H}\prod_{j=1}^{W}\mathrm{P}(\boldsymbol{S}_{\mathrm{I}i,j}|\boldsymbol{S}_\mathrm{C}, \boldsymbol{B}) = \prod_{i=1}^{H}\prod_{j=1}^{W}\mathrm{P}(\boldsymbol{S}_{\mathrm{I}i,j}|\boldsymbol{z}_k) \tag{7}$$

where $\boldsymbol{z}_k$ denotes the image representations [26] and $k = \lfloor \frac{iW+j}{T^2} \rfloor$. To better optimize the deep network, the entire network uses a residual connection and layer normalization (Add & Norm).

3.2. Radical-and-transformer-based recognition network

We employ the framework of RTN-R [9] as recognizing network, which is symmetrical to the generator and has the similar encoder and decoder with the generator's. The model is composed of a convolutional feature extractor (CNN), Transformer encoder and decoder with L layers, which are outlined in Table 1.

The model first learns to encode the source images $\boldsymbol{S}_\mathrm{I}$ into high-level representations $\boldsymbol{Z}$. PE is added on the character representations $\boldsymbol{Z}$. Then the encoder captures information from the entire sequence to build context vector $\boldsymbol{C}$. Finally, the decoder uses this context vector to generate variable-length output sequence $\boldsymbol{X}$ word by word, which is radical representations. At each step the model is auto-regressive, consuming the summation of previously generated symbols' representations and TPE as additional input when generating the next. Given a source image $\boldsymbol{S}_\mathrm{I}$ and a target caption $\boldsymbol{S}_\mathrm{C}$, the recognizer can be defined as the following conditional probability:

$$\mathrm{P}(\boldsymbol{S}_\mathrm{C}|\boldsymbol{S}_\mathrm{I}) = \prod_{i=1}^{|\boldsymbol{S}_\mathrm{I}|}\mathrm{P}(\boldsymbol{S}_{\mathrm{C}i}|\boldsymbol{S}_{\mathrm{C}<i}, \boldsymbol{S}_\mathrm{I}) = \prod_{i=1}^{|\boldsymbol{S}_\mathrm{I}|}\mathrm{P}(\boldsymbol{S}_{\mathrm{C}i}|\boldsymbol{c}_i) \tag{8}$$

where $\boldsymbol{S}_{\mathrm{C}<i}$ denotes a prefix of $\boldsymbol{S}_\mathrm{C}$ with length $i-1$.

Similar to generation tasks, we define the multiplication of attention matrices as visual attention $\boldsymbol{A}_\mathrm{r}$, which can choose the most relevant part from the whole input image for calculating the context vector.

3.3. Joint optimization

Joint training can help the generator and recognizer learn from each other effectively [27]. In this section, we introduce the two sub-tasks of joint optimization and how they help the generator and recognizer:

3.3.1. Contrastive learning

Considering the symmetry of recognizer and generator, we define the output of generator decoder and the input of recognizer encoder as a set of samples, which are both the representations of characters. Given the character representation of generator, the model needs to identify the true character representation of recognizer in a set of candidate representations. Similarly, we define the input of generator encoder and the output of recognizer decoder as a set of representations of radicals, and we also employ contrastive learning on radical representations.

For recognizer, input images are transformed to character representations $\boldsymbol{Z}$, then radical representations $\boldsymbol{X}$ and finally radical symbols. Meanwhile, the generator processes the symbols in a reverse way. Considering the symmetry of generator and recognizer, we propose three sets of contrastive learning on radical representations, character representations and attention matrices.

- Radical and character representations: For each Chinese character, the radical presentation $\boldsymbol{X}$ in generation and recognition should be close to each other in the mapped space. We employ learnable matrices $\boldsymbol{W}^{\mathrm{R}}$ to map radicals to contrastive learning space as $\boldsymbol{r}$.

$$\begin{aligned} \boldsymbol{r}_{\mathrm{g}} &= \boldsymbol{W}^{\mathrm{R}}_{\mathrm{g}}\boldsymbol{X}_{\mathrm{g}} \\ \boldsymbol{r}_{\mathrm{r}} &= \boldsymbol{W}^{\mathrm{R}}_{\mathrm{r}}\boldsymbol{X}_{\mathrm{r}} \end{aligned} \tag{9}$$

  where $\boldsymbol{W}^{\mathrm{R}} \in \mathbb{R}^{D \cdot N \times D}$ and $\boldsymbol{r} \in \mathbb{R}^{D}$, subscript g and r represent whether the variable belongs to generation or recognition. Similarly, for character representations $\boldsymbol{Z}$, we employ $\boldsymbol{W}^{\mathrm{C}}$ to map $\boldsymbol{Z}$ to contrastive space as $\boldsymbol{c}_{\mathrm{g}}$ and $\boldsymbol{c}_{\mathrm{c}}$. By imposing the vector constraint on radical and character presentations, the generator and recognizer can learn from each other and jointly improve the performance of the network.
- Attention matrices: The generator and recognizer learn textual attention $\boldsymbol{A}_{\mathrm{g}}$ and visual attention $\boldsymbol{A}_{\mathrm{r}}$ during training and build the correspondence between radicals and zones in both directions. Given the symmetry of the visual attention and textual attention, we utilize mean-square errors as the contrastive loss for evaluating the similarity between the two attention matrices.

3.3.2. Dual learning

Noticing that generator and recognizer can always happen in dual directions, we employ them as evaluators for each other. Starting from source images $\boldsymbol{S}_{\mathrm{I}}$, we first feed it into the recognizer $\mathcal{R}$ and then further feed the prediction results $\boldsymbol{P}$ into the generator $\mathcal{G}$. By evaluating the output image $\boldsymbol{F}$, we will get a sense of the quality of the two models, and be able to improve them accordingly. The process starting from source captions $\boldsymbol{S}_{\mathrm{C}}$ is the same as above.

3.4. Optimization strategy

During training, the generator and recognizer learn to model Chinese characters by solving generation loss $\mathcal{L}_{\mathrm{g}}$ and recognition loss $\mathcal{L}_{\mathrm{r}}$. This is augmented by joint losses $\mathcal{L}_{\mathrm{j}}$ to encourage the generator and recognizer to learn from each other.

$$\mathcal{L} = \mathcal{L}_{\mathrm{g}} + \mathcal{L}_{\mathrm{r}} + \alpha\mathcal{L}_{\mathrm{j}} \tag{10}$$

where α is a tuned hyper-parameter.

3.4.1. Generation loss

$\mathcal{L}_{\mathrm{g}}$ is employed to guide the character generation. We use mean-square error (MSE) as the loss function to measure the Euclidean distance between output image $\boldsymbol{F}$ and target image $\boldsymbol{S}_{\mathrm{I}}$. The generator learns coding of glyph by minimizing $\mathcal{L}_{\mathrm{g}}$.

$$\mathcal{L}_{\mathrm{g}} = \|\boldsymbol{F} - \boldsymbol{S}_{\mathrm{I}}\|_2 \tag{11}$$

3.4.2. Recognition loss

We use cross-entropy (CE) as the loss function that the recognizer learns pixel-radical correspondence.

$$\mathcal{L}_{\mathrm{r}} = -\boldsymbol{S}_{\mathrm{C}} \cdot \log(\boldsymbol{P}) \tag{12}$$

where $\boldsymbol{P}$ represents the prediction results of recognizer, $\boldsymbol{S}_{\mathrm{C}}$ represents the target caption.

3.4.3. Joint loss

The generation and recognition tasks build the two models independently, and the joint loss $\mathcal{L}_{\mathrm{j}}$ is designed to increase the association of them. The joint loss is a summation of two terms: contrastive loss $\mathcal{L}_{\mathrm{c}}$ and dual loss $\mathcal{L}_{\mathrm{d}}$.

$$\mathcal{L}_{\mathrm{j}} = \mathcal{L}_{\mathrm{c}} + \beta\mathcal{L}_{\mathrm{d}} \tag{13}$$

where β is a tuned hyper-parameter.

For contrastive loss $\mathcal{L}_{\mathrm{c}}$, we employ InfoNCE loss [28] to measure the similarity of radical representations $\boldsymbol{R}$ and character representations $\boldsymbol{C}$ in contrastive space, and employ MSE loss to measure the Euclidean distance between attention matrices $\boldsymbol{A}_{\mathrm{g}}$ and $\boldsymbol{A}_{\mathrm{r}}$. Given radical level representations $\boldsymbol{r}_{\mathrm{g}}$ and character level representations $\boldsymbol{c}_{\mathrm{g}}$ of generator, the model needs to identify the true representations of recognizer $\boldsymbol{r}_{\mathrm{r}}$ and $\boldsymbol{c}_{\mathrm{r}}$ in a set of candidate representations $\tilde{\boldsymbol{r}} \in \boldsymbol{D}_{\mathrm{r}}$ and $\tilde{\boldsymbol{c}} \in \boldsymbol{D}_{\mathrm{c}}$, which include the true representations and M distractors. Distractors are uniformly sampled from other samples of the same batch. The loss is defined as:

$$\begin{aligned} \mathcal{L}_{\mathrm{c}} = &-\gamma_{\mathrm{r}} \cdot \log \frac{\exp(\mathrm{sim}(r_{\mathrm{g}}, r_{\mathrm{r}})/\tau)}{\sum_{\tilde{\boldsymbol{r}} \sim \boldsymbol{D}_{\mathrm{r}}} \exp(\mathrm{sim}(r_{\mathrm{g}}, \tilde{\boldsymbol{r}})/\tau)} \\ &-\gamma_{c} \cdot \log \frac{\exp(\mathrm{sim}(c_{\mathrm{g}}, c_{\mathrm{r}})/\tau)}{\sum_{\tilde{\boldsymbol{c}} \sim \boldsymbol{D}_{\mathrm{c}}} \exp(\mathrm{sim}(c_{\mathrm{g}}, \tilde{\boldsymbol{c}})/\tau)} \\ &-\gamma_{a} \cdot \log\left(1 - \frac{\|\boldsymbol{A}_{\mathrm{r}}^{\top} - \boldsymbol{A}_{\mathrm{g}}\|_2}{M \cdot N \cdot N}\right) \end{aligned} \tag{14}$$

where $\boldsymbol{A}_{\mathrm{g}}$ and $\boldsymbol{A}_{r}$ represent the textual and visual attention, M represents the number of attention head, $M \cdot N \cdot N$ represents the size of attention matix and γ are tuned hyper-parameters. We compute the cosine similarity $\mathrm{sim}(\boldsymbol{a}, \boldsymbol{b}) = \boldsymbol{a}^{\top}\boldsymbol{b}/\|\boldsymbol{a}\|_2\|\boldsymbol{b}\|_2$ for radical representations and character representations. Dual loss $\mathcal{L}_{\mathrm{D}}$ is defined as:

$$\mathcal{L}_{\mathrm{d}} = \|\mathcal{G}(\boldsymbol{P}) - \boldsymbol{S}_{\mathrm{I}}\|_2 - \boldsymbol{S}_{\mathrm{C}} \cdot \log(\mathcal{R}(\boldsymbol{F})) \tag{15}$$

where $\mathcal{G}$ and $\mathcal{R}$ represent the function of the generator and recognizer.

4. Tree position embedding

Since our model contains no recurrence, we must inject some information about the relative or absolute position of the tokens. We add position embedding to each sequence representations to distinguish the order of tokens, which is proven effective in the published works. Traditional position embedding (PE) [23] is defined as:

$$\begin{aligned} \mathrm{PE}_{i,2d} &= \sin\left(\frac{i}{\lambda^{2d/D}}\right) \\ \mathrm{PE}_{i,2d+1} &= \cos\left(\frac{i}{\lambda^{2d/D}}\right) \end{aligned} \tag{16}$$

where i is the position, d is the dimension and λ is set to 10,000.

The distances of 1D position embedding are determined by the distance of the tokens in the sequence. However, considering the difference between sequence structure and tree structure, PE is not quite applicable here for tree structure captions. Figure 5 shows the way of node and layer numbering in a tree, and we elaborate why PE is not suitable for tree structure information through it. Obviously, the position embedding of tree structure should satisfy two constraints: (1) The embedding of child nodes should be at the same distance from the one of parent node. (2) The embedding of nodes should be closer if they appear closer in a tree. However, PE will result in erroneous conclusions. Each parent node and its child nodes do not comply with the 1st constraint. Moreover, since the nodes follow the depth-first traversal as a sequence, some nodes belong to different sub-trees in a tree, but they are adjacent in a

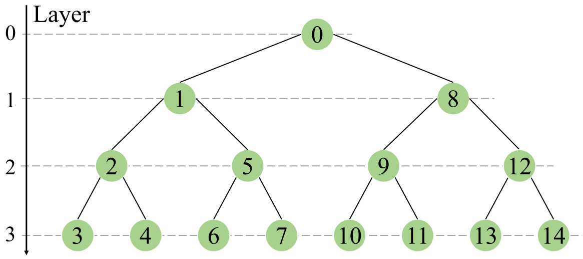


Fig. 5. The number of tree nodes is shown. The tree is recorded as a sequence arranged according to depth-first traversal from 0 to 14.

sequence. For example, in Fig. 5 node 7 is adjacent to node 6 and 8 in the sequence, but node 8 is much too far away from node 7 in the tree.

Thus, we propose a novel tree position embedding (TPE), which complies with the above constraints. The model takes the tokens following the depth-first traversal as a sequence. TPE can help our model distinguish the position of each token. We use sine functions of different frequencies to represent root node, left and right child node in each layer.

$$\left.\begin{array}{r} Root = \sin\left(\frac{2d\pi}{D}\right) \\ L_l = \sin\left(\frac{(4l-2)d\pi}{D}\right) \\ R_l = \sin\left(\frac{4ld\pi}{D}\right) \end{array}\right| \quad d = 1, \ldots, D \tag{17}$$

where l represents the layer, d is the dimension, L_l and R_l represent the left and right child node in lth layer. That is, each node corresponds to a sinusoid and the periods are restricted to integer.

The tree position embedding $\mathrm{TPE}(i) \in \mathbb{R}^D$ of position i is defined as a mean of the representations of i and all its recurrent parent nodes, which can be expressed as the following equations:

$$\begin{aligned} &\mathrm{TPE}(1) = (Root + L_1)/2 \\ &\mathrm{TPE}(4) = (Root + L_1 + L_2 + R_3)/4 \end{aligned} \tag{18}$$

We choose this function because we hypothesized it would allow the model to easily learn to observe relative positions in a tree structure. TPE can better describe the positional relevance between tree nodes, which can be mathematically proved. We denote $\phi_f(\cdot,\cdot)$ as a function to calculate closeness/proximity between embedded positions, which is used in the later proof. Equations ((17), (18)) can be reduced to the following property.

$$\phi_f(x, y) = f(x)^{\mathrm{T}} \cdot f(y) \tag{19}$$

4.1. Equality

Intuitively, child nodes $2i+1$ and $2i+2$ should be at the same distance from parent node i. PE will result in erroneous conclusions obviously.

$$\begin{aligned} &\phi_{\mathrm{PE}}(i, 2i+1) - \phi_{\mathrm{PE}}(i, 2i+2) \\ &= \sum_{d=1}^{D/2}\left[\cos\left(\frac{i+1}{\lambda^{2d/D}}\right) - \cos\left(\frac{i+2}{\lambda^{2d/D}}\right)\right] \end{aligned} \tag{20}$$

However, TPE fulfils the above-mentioned property. We define $l = \lfloor \log_2(i+1) \rfloor + 1$ as the layer of node i.

$$\phi_{\mathrm{TPE}}(i, 2i+1) - \phi_{\mathrm{TPE}}(i, 2i+2) = \frac{l}{l+1} - \frac{l}{l+1} = 0 \tag{21}$$

In addition, since $Root$, L_l and R_l are orthogonal to each other, the child nodes having the same parent node can be separated from one another. In other words, although child nodes can be at the same PE distance from parent node by giving the child nodes the same numbering, the model can not distinguish child nodes.

$$\phi_{\mathrm{TPE}}(2i+1, 2i+2) = \frac{l^2}{(l+1)^2} \neq 1 \tag{22}$$

4.2. Location relevance

Generally, if words appear close to each other in a tree, they are more likely to determine the semantics together. Taking node i as an example, it should be closer to its child node $2i+1$ than node n, where $n \notin \{i, 2i+1, 2i+2, \lfloor \frac{i-1}{2} \rfloor\}$.

$$\phi_{\mathrm{TPE}}(i, 2i+1) - \phi_{\mathrm{TPE}}(i, n) \geq \frac{l}{l+1} - \frac{(l-1)^2}{l^2} > 0 \tag{23}$$

The results indicate that TPE can establish correct correspondence of positions in a tree structure, while PE cannot. Only when $n > 2i+2$, PE is eligible.

$$\phi_{\mathrm{PE}}(i, 2i+1) - \phi_{\mathrm{PE}}(i, n) = \sum\nolimits_{d=1}^{D/2}\left[\cos\left(\frac{i+1}{\lambda^{2d/D}}\right) - \cos\left(\frac{n-i}{\lambda^{2d/D}}\right)\right] \tag{24}$$

5. Experimental setup

5.1. Datasets

We prepare the printed and natural scene characters and design related experiments to verify the effectiveness of our method in simple and complex scenes.

- **Printed Text**: In the experiments on printed characters, we choose 27,533 Chinese characters in 8 font styles (i.e. Hei, Song, Kai, Deng, Fangsong, Li, Yao, Youyuan) as database and divide them into training set and testing. There are 180,052 printed characters in 8 font styles in training set, which contains 22,933 classes. And there are 31,760 printed characters in testing set, which contains 4172 classes that do not appear in training set.
- **Scene Text**: The natural scene database is the Chinese Text in the Wild (CTW) [29], which includes 1,018,402 Chinese characters and 6 types of character attributes for each one: occluded, complex background, distorted, 3D raised, WordArt and handwritten. Following the official dataset splitting, we use 3580 Chinese character categories with 760,107 instances for training, 2015 Chinese character categories with 52,765 instances for validation and 103,519 instances for testing.
- **Handwriting Text**: We select the ICDAR-2013 competition [30] of HCCR and CASIA [31] database as handwriting database. For training, we adopt the CASIA database including HWDB 1.0 and 1.1. The ICDAR-2013 database is used to evaluate the performance on within-dictionary Chinese characters, and CASIA HWDB 1.2 is used to evaluate the performance on out-of-dictionary Chinese characters. There are 3319 non-common Chinese characters in HWDB1.2 dataset and we pick 3277 classes, which is the same as that in DenseRAN [32]. Note that the Chinese characters in HWDB1.2 dataset are not appeared in training set.
- **Random Caption**: The output images of generator depend on the basic of input captions, which describe the details of characters. Therefore, we prepare 3000 random caption of novel/nonexistent Chinese characters as a validation set to evaluate the generalization performance of the generator.

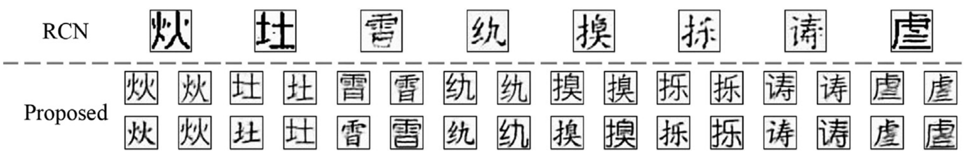


Fig. 6. We employ 9 novel characters to evaluate the generalization of our model. The first row gives the example results of RCN [3]; the last row gives the example results of the proposed attention-based generator.

5.2. Settings

The images for training and testing are resized as 32 × 32. The captions contain 624 radicals and 10 spatial structures for Chinese characters. The parameters of Transformer blocks are illustrated in Table 1. The model dimension D is 512. The up-sampling module of generator has seven blocks and the temporal convolutions with strides (2,1,2,1,1,2,1) and kernel widths (3,3,3,3,3,5,5). We employ a 16-layer VGG network [33] as feature extractor in recognizer. The length of background feature K is 16, where H, W and T are 32, 32 and 8. The hyper-parameters α, β, γ_r, γ_c and γ_a are 0.5, 0.2, 1, 1 and 0.8, which are tuned according to the validation set. Each task uses dropout rate 0.2. The temperature τ in the contrastive loss is set to 0.1.

Besides, the models were implemented with PyTorch 1.3. In the experiments on printed scene, batches are built by 32 images and we train on a single V100 32G GPU for 5 h. In the experiments on natural scene and handwriting scene, batches are built by 256 images and we train on 4 V100 32G GPU for 30 h and 18 h respectively.

6. Experiments

In this section, we will show the effectiveness of the proposed joint optimization. From Sections 6.1 to 6.3, we tested our model in printed scene, natural scene and handwriting scene respectively. The experiments are designed to answer the three questions: **Q1** Are contrastive learning, dual learning and TPE effective in the joint optimization; **Q2** Compared with other state-of-the-art methods, does our model perform better than other models; **Q3** Can joint optimization outperform traditional data augmentation?

6.1. Experiments on printed text (Q1)

Evaluating the quality of synthesized images is an open and difficult problem. So the experiments are first performed on simple scenarios (printed dataset). For generation, mean-square error (MSE) is used to evaluate the differences between the target images and generated images, and structural similarity (SSIM) is used to measure the similarity of two images. Top1 and Top5 accuracy rates are employed as the indicators of recognizer. We summarize the overall performances in Table 2.

Table 2
Ablation of joint optimization in printed scene. **SIM**: contrastive learning. **DUAL**: dual learning. **TPE/PE**: tree or 1D position embedding is adopted. MSE: mean-square error. SSIM: structural similarity. Top1/Top5: recognition accuracy rate (%).

	SIM	DUAL	TPE/PE	MSE	SSIM	Top1	Top5
Isolated				0.232	0.79	85.17	90.88
			PE	0.211	0.82	86.55	96.64
			TPE	0.199	0.851	87.22	96.81
Joint	✓		TPE	0.172	0.887	90.11	97.53
		✓	TPE	0.178	0.815	89.20	97.3
	✓	✓	PE	0.166	0.913	92.55	98.14
	✓	✓	TPE	**0.160**	**0.929**	**93.61**	**98.4**

We design an ablation experiment to verify the validity of the sub-tasks in joint optimization and the effectiveness of TPE. In Table 2, "isolated" represents the generator and recognizer optimize independently; "joint" represents the joint optimization tasks are primed. Table 2 clearly declares that joint optimization improves the performance of both generator (MSE reduced from 0.199 to 0.160, SSIM increase from 0.851 to 0.929) and recognizer (Top1 accuracy increased from 87.22% to 93.61%, Top5 accuracy increased from 96.81% to 98.4%). The first 2 lines in results of joint model indicate more significant contributions of contrastive learning than dual learning. The last 2 lines in results of joint model imply that TPE plays a more important role than PE.

To check the generalization of the model, random captions are used here. As illustrated in Fig. 6, we compare the generating images of RTN-G and RCN [3]. Evidently, the proposed RTN-G using joint optimization and TPE performs better. More specifically, in Fig. 7, we show how the generator learns to produce an unseen Chinese character. The generator finds the best match from the attention map and well generates the alignment strongly corresponding to human intuition.

6.2. Experiments on scene text

In this section, we evaluate our model in a natural scene database. Compared with the printed Chinese character dataset, CTW dataset is more challenging due to its diversity and complexity. All CTW data are used for recognition tasks, and part of CTW data are adopted for generation and joint optimization tasks.

6.2.1. Ablation experiments on joint optimization (Q1)

We employ the same method as in the Section 6.1 to evaluate the performance of our model in natural scenes on CTW dataset. According to the officially provided annotations, the low-quality samples were removed from the training set of the generation tasks. Since the generator requires character attributes, the characters with complex backgrounds and artistic fonts can not be created by the current version. We employ characters with "distorted" and "raised" attributes for generation and joint optimization tasks only. For character attributes, we acquire the approximation of background and text color by pixel statistics and mathematical operation. A pre-trained classification network is employed to identify the font styles.

Figure 8 shows the generated results in natural scenes, where the left part shows the source image and the middle part shows the generated characters. In the right part of Fig. 8, we synthesize a seemingly realistic photograph of streets-capes by sticking these generated characters back to the source image. Although the current version can only operate in simple artificial scenes, the synthesized images are entirely realistic-looking.

As for recognition tasks, we employ the results in [9] as a baseline. We illustrate the performance in Table 3, compared with baseline (Top1 accuracy: 87.31%), TPE and joint optimization provide an additional boost in recognition performance (+1.33%). Moreover, TPE improves the recognition rate by 0.13% and 0.42% over PE in isolated model and joint model. Limited to the capacity of generator, only partial data are adopted for the joint optimization which makes recognition rate hard to vastly promote. With

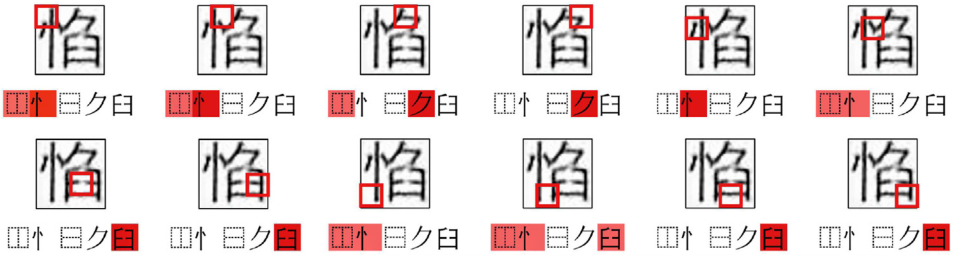


Fig. 7. Attention visualization of generating an unseen Chinese character.

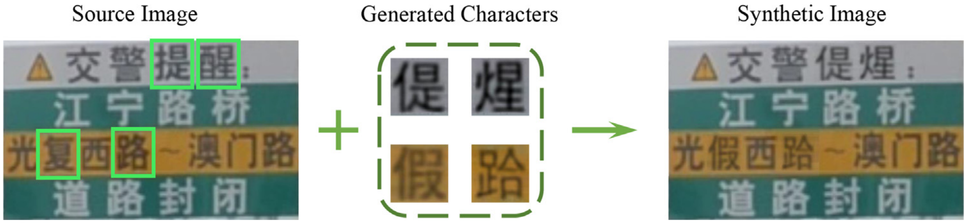


Fig. 8. Exampling results of our model in nature scenes on CTW dataset. The left part gives the original image. The middle part gives the generated characters which have the same attributes as the original image. According to the coordinate of each character, we replace the character in the original image and obtain a synthetic image as shown in the right part.

Table 3
Ablation of joint optimization in natural scene on CTW dataset. **SIM**: contrastive learning. **DUAL**: dual learning. **TPE/PE**: tree or 1D position embedding is adopted. MSE: mean-square error. SSIM: structural similarity. Top1/Top5: recognition accuracy rate (%).

	SIM	DUAL	TPE/PE	MSE	SSIM	Top1	Top5
Isolated				0.284	0.625	86.14	90.76
			PE	0.263	0.699	87.31[9]	91.54
			TPE	0.244	0.723	87.44	91.63
Joint	✓		TPE	0.229	0.765	87.60	91.73
		✓	TPE	0.223	0.771	87.91	92.02
	✓	✓	PE	0.214	0.82	88.22	92.15
	✓	✓	TPE	**0.199**	**0.858**	**88.64**	**92.44**

Table 4
The generation results of SynthText, SRNet and our model. Compared with character based synthesis algorithms, radical based generation method can create out-of-dictionary characters. Meanwhile, the qualities of within-dictionary characters are very similar.

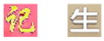
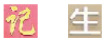
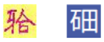
	Within-dictionary	Out-of-dictionary
SynthText [34]		
SRNet [35]		
Ours		

Table 5
Evaluation of character generation and recognition systems on CTW. MSE: mean-square error. SSIM: structural similarity. Top1/Top5: radical expression recognition accuracy rate (%). FPS: frames per second in testing processing.

	MSE	SSIM	FPS	Top1(%)	Top5(%)	FPS
ResNet50 [29]				79.00	85.9	212.8
DenseNet [36]				79.45	86.3	323.7
RAN [36]				85.22	90.15	86.3
RTN-R [9]				87.31	91.54	92.2
SynthText [34]	0.325	0.521	894.4			
RCN [3]	0.311	0.558	13.5			
SRNet [35]	0.212	0.788	90.6			
RTN-G	0.244	0.723	141.7			
RTN-R+RTN-G	**0.199**	**0.838**	141.7	**88.64**	**92.44**	92.2

the improvement of generator in the future, the recognizer should benefit more from the joint optimization.

6.2.2. Comparison with state-of-the-arts (Q2)

We compare our model with other state-of-the-art generation and recognition methods to show the competitive results of our proposed joint model. In order to examine the robustness of the joint model on generation tasks, we compare our model with SynthText[34] and SRNet [35]. In fact, SynthText and SRNet are not fully comparable with our model. SynthText needs True Type Font files as input and SRNet needs character image as input which focuses on style transfer. However, the input of our model is just text that includes less information. To make it fair, the results are generated by CTW dataset without any additional data. Besides, since the CTW dataset does not contain all Chinese characters, we split all Chinese characters into within-dictionary characters and out-of-dictionary characters.

As shown in Table 4, we prepare 2 within-dictionary characters and 2 out-of-dictionary to evaluate the performance. The quality of our model performs similarly to other generation methods, even the input information of our model is relatively weak. Compared with other generation models, our model can create out-of-dictionary characters, while SynthText and SRNet can never create novel characters.

In experiments on recognition, except that the experimental results of our model are measured by the tool provided by CTW, all other experimental results are selected from published papers. For fairness, all our experiments only use offline information in the official CTW training set without additional data augmentation. As shown in the Table 5, we select mainstream radical and attention based generation and recognition algorithms for comparison. Obviously, joint optimization model (RTN-G + RTN-R) is significantly better than other single models in terms of character generation and recognition. Among those, "RAN" (Radical Analysis Network), "RTN-R" are advanced radical-based recognition models in recent years. These experimental results fully prove that joint optimization model has good generalization and superior performance for Chinese character generation and recognition. In addition, we display the time cost of each model in testing processing. In these recognition models, ResNet50 and DenseNet are character-based models, which do not need decode radical sequences. They cost

Table 6
Evaluation of recognition models on CTW with respect to 6 attributes; "all" includes all characters on the test dataset.

	RAN	RTN-R	RTN-R+RTN-G
occluded	71.55	**73.94**	72.64
background	82.84	84.54	**85.13**
distorted	71.55	83.60	**83.66**
3D raised	76.17	78.06	**78.92**
wordart	**87.11**	84.25	86.95
handwritten	63.58	66.70	**67.27**
all	85.56	87.31	**88.64**

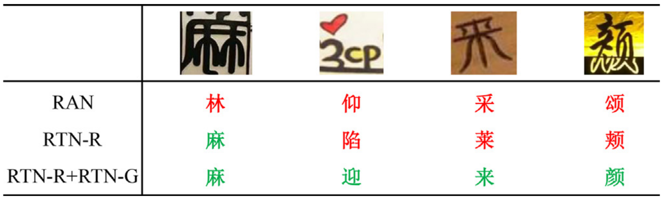

RAN
RTN-R
RTN-R+RTN-G
林
仰
采
颂
麻
陷
莱
频
麻
迎
来
颜

Fig. 9. The example comparison among RAN, RTN-R and joint model in recognition tasks. The red characters denote the error predicted result and the green characters denote the correct ones.

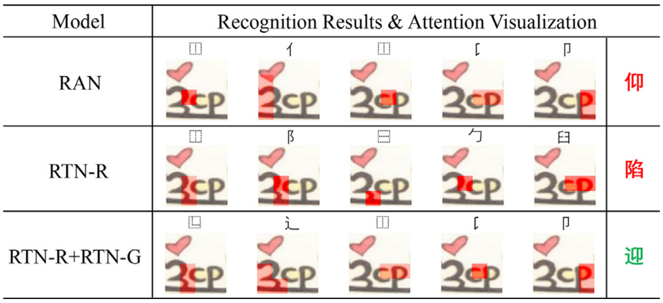

Model
Recognition Results & Attention Visualization
RAN
⿰
亻
⿰
卩
仰
RTN-R
⿰
阝
⿱
勹
臼
陷
RTN-R+RTN-G
⿺
辶
⿰
卩
迎

Fig. 10. The comparison of recognition results and attention visualization among RAN, RTN-R and RTN-R+RTN-G in recognition tasks. The joint model corrects error predicted results of single model, and the attention becomes more accurate after joint optimization.

less than radical-based models RAN and RTN-R. The generation model SynthText is a rule-based non-neural method, so its speed is much higher than others. Moreover, our joint model does not need additional time compared to a single model (RTN-R or RTN-G) in testing processing.

To demonstrate the robustness of our model in natural scenes, Table 6 shows the performance of recognition models with respect to 6 attributes: occluded, complex background, distorted, 3D raised, wordart characters and handwritten characters. We can observe that the recognition rates of joint model are much higher than others in complex background, distorted, 3D raised and handwritten. However, our joint model performs worse than RTN-R in occluded, since the generator cannot create a high-quality image with occlusion, which brings negative effect to the recognizer. As for wordart, our joint model achieves higher accuracy rate than single model RTN-R, but still no more than RAN.

In Fig. 9, we compare the recognition results of RAN, RTN-R and our proposed joint model. Obviously, complex samples with artistic fonts are challenging to process for single recognition model. However, joint optimization can address this problem well. In addition, Fig. 10 shows the recognition results and attention visualization of a complex sample. The joint model corrects error predicted results of single model and attention becomes more in line with the work mechanism of human vision. The performance of feature extraction and feature classification of recognition model improves with the help of generation model.

Table 7
Comparison of accuracy rate (%) among offline augmentation and online augmentation of within-dictionary characters and out-of-dictionary characters on CTW dataset.

	Additional Data	Augmentation Type	Top1	Top5
RTN-R	–	–	87.44	91.63
	ADS_Synth	Offline	87.32	91.54
	ADS_SRNet	Offline	87.68	91.98
	ADS_Ours (Within)	Offline	87.56	91.76
	ADS_Ours (Out)	Offline	**88.42**	**92.31**
RTN-G+RTN-R	–	–	88.64	92.47
	Text_Within	Online	88.72	92.88
	Text_Out	Online	**88.91**	**94.21**

6.2.3. Comparison with data augmentation (Q3)

Traditional data augmentation aims to use generation models to create synthetic images and employ them as training set on recognition tasks. We define traditional data augmentation as offline augmentation since the generator and recognizer are independent. However, our proposed generator and recognizer are not. The joint model can provide not only synthetic images but also information of generation. We define this method as online augmentation, which needs generator and recognizer to run at the same time. We believe that the cost of online augmentation is less than offline augmentation, and it performs better. In addition, traditional generation methods can not create an out-of-dictionary Chinese character, which is an advantage of our model. Since the number of radical combinations influence the performance directly and out-of-dictionary Chinese character contains a large number of novel radical combinations, our method can provide more valuable dataset. To further assess the impact of online/offline augmentation and within/out-of-dictionary characters on recognition tasks, we conduct recognition experiments on RTN-R.

We first use within-dictionary character set (Text_Within) to build additional data sets ADS_Synth, ADS_SRNet and ADS_Ours (Within) by SynthText, SRNet and our model. The above three data sets are used to compare the effect of with-in-dictionary data augmentation on recognition performance. Then, we build ADS_Ours (Out) with out-of-dictionary character set (Text_Out), including existed characters and random captions (mentioned in Section 5.1). Each additional dataset contains 5 million images.

As shown in Table 7, offline augmentation of within-dictionary characters improves recognition rate up to only 0.24% in large data sets. Since offline augmentation can only provide seen characters with different backgrounds and CTW is of enough variety, the diversity brought by offline augmentation of within-dictionary characters is minimal. Evidently, offline augmentation of out-of-dictionary characters is more effective and improves recognition rate up to 0.98%. Novel radical combinations of training data makes the recognizer more robust and generalized. The results of online augmentation also verify this conclusion. Moreover, ADS_Synth, ADS_SRNet and ADS_Ours(Within) are built from the same character set, and they are comparable. It is worth noting that SRNet achieves the highest image quality metric among single models in Table 5, and ADS_SRNet contributes the most to the recognition model in the above 3 additional datasets. This illustrates that higher-quality images improve the recognition rate.

Furthermore, the results also demonstrate the great superiority of online augmentation, the accuracy rate achieves 88.91%. Compared with offline augmentation, more information during generation is used by recognizer instead of a synthetic image only. Contrastive learning and dual learning improve the performance of feature extraction and feature classification of recognition model. Moreover, the storage requirement and reading/writing operation of online augmentation is lower than offline augmentation, since

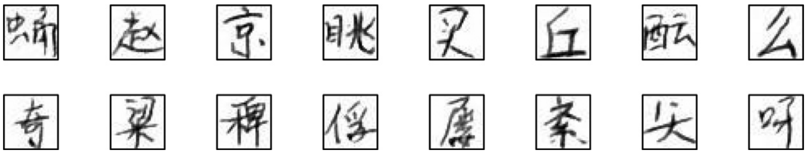


Fig. 11. The generated results of our model in handwriting scene.

Table 8
Performance comparison of our method and other state-of-the-art methods on ICDAR-2013 competition set and HWDB1.2 set. FPS: frames per second in testing processing.

	ICDAR-2013	HWDB1.2	FPS
Direct+ConvNet [37]	96.13	0	522.4
DenseNet [32]	95.9	0	323.7
VGG14-RAN [32]	93.79	38.74	86.3
Dense-RAN [32]	96.66	40.82	76.5
TAN [11]	96.78	42.88	52.1
RTN-R	96.72	42.53	92.2
RTN-R+RTN-G	**96.88**	**44.12**	92.2

online augmentation is synchronous and offline augmentation is asynchronous.

6.3. Experiments on handwriting text (Q2)

In order to prove that our method still performs well in handwriting scene, we compare our model with other state-of-the-art recognition methods. We employ CASIA HWDB1.0 and 1.1 as training set, which contains 2,674,784 samples. The testing set includes within-dictionary character set (ICDAR-2013) and out-of-dictionary character set (HWDB1.2), since radical-based recognizer and generator can handle out-of-dictionary characters, while character-based model cannot.

We list the state-of-the-art recognition methods of HCCR in Table 8. VGG14-RAN and Dense-RAN are RAN models with VGG14 encoder and DenseNet encoder, respectively. TAN (Tree-structure Analysis Network) is an improved radical-based recognition model, which introduces a tree decoder and makes full use of the information of tree structure in a Chinese character. The recognition rate of TAN is superior to other single recognition methods due to powerful tree decoder. The joint optimization improves the accuracy rate of RTN-R and exceeds TAN, which achieves 96.88% on within-dictionary character set and 44.12% on out-of-dictionary character set. The recognizer benefits from the knowledge of generator and performs better than a single recognition model. Multi-level information is shared between generator and recognizer during the joint optimization.

Moreover, Fig. 11 shows the generated results of RTN-G. We can observe that despite our generator cannot work well enough on complex Chinese characters with a large number of strokes, most characters look real.

7. Conclusion

In this paper, we introduce a novel attention-based Chinese character generator, a joint optimization mechanism and tree position embedding. Compared with the recursive generator, our proposed model can generate Chinese characters in natural scene, and performs better on out-of-dictionary Chinese characters. The tree position embedding captures the hierarchical structure of Chinese characters and enhances our model's ability to generate and recognize complex characters. Experimental results show that our proposed model outperforms the state-of-the-art baseline model on both character generation and recognition tasks. The joint optimization mechanism and tree position embedding greatly contribute to the improvement of our model's performance.

Furthermore, we aim to explore the generalizability of our joint optimization strategy to other tasks that involve symmetrical model systems. In future work, we plan to make efforts in generating Chinese characters on line level to achieve joint optimization of text generation and recognition. Considering the effect of our generated images in natural scene, we plan to discuss the erasure of characters in radical-level, which also is a valuable research. In addition, we will improve our model to handle the noise/occlusion/distortion problems and design corresponding optimization method for such problems. Finally, the reduction of computational cost is also our concern.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. Wichmann, W. Brendel, ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness, arXiv:abs/1811.12231(2019).
[2] Y. Zhu, J. Du, J. Zhang, Dual learning of the generator and recognizer for chinese characters, in: 2017 4th IAPR Asian Conference on Pattern Recognition (ACPR), IEEE, 2017, pp. 536–541.
[3] M. Xue, J. Du, J. Zhang, Z.-R. Wang, B. Wang, B. Ren, Radical composition network for chinese character generation, in: Document Analysis and Recognition – ICDAR 2021, Springer International Publishing, 2021, pp. 252–267.
[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.
[5] Y. Wang, Y. Xia, T. He, F. Tian, T. Qin, C. Zhai, T.-Y. Liu, Multi-agent dual learning, in: Proceedings of the International Conference on Learning Representations (ICLR) 2019, 2019.
[6] A.-B. Wang, K.-C. Fan, Optical recognition of handwritten chinese characters by hierarchical radical matching method, Pattern Recognit. 34 (1) (2001) 15–35.
[7] D. Shi, R.I. Damper, S.R. Gunn, Off-line handwritten chinese character recognition by radical decomposition, ACM Trans. Asian Lang. Inf.Process. 2 (1) (2003) 27–48.
[8] C. Dong, J. Zhang, C. Zong, M. Hattori, H. Di, Character-based LSTM-CRF with radical-level features for chinese named entity recognition, in: Natural Language Understanding and Intelligent Applications, Springer, 2016, pp. 239–250.
[9] C. Yang, Q. Wang, J. Du, J. Zhang, C. Wu, J. Wang, A transformer-based radical analysis network for chinese character recognition, in: 2020 25th International Conference on Pattern Recognition (ICPR), IEEE, 2021, pp. 3714–3719.
[10] J. Zhang, Y. Zhu, J. Du, L. Dai, Radical analysis network for zero-shot learning in printed chinese character recognition, in: 2018 IEEE International Conference on Multimedia and Expo (ICME), IEEE, 2018, pp. 1–6.
[11] Y. Li, J. Du, J. Zhang, C. Wu, A tree-structure analysis network on handwritten chinese character error correction, IEEE Trans. Multimed. (2022), doi:10.1109/TMM.2022.3163517. 1–1
[12] Q. Lin, L. Liang, Y. Huang, L. Jin, Learning to generate realistic scene chinese character images by multitask coupled GAN, in: Chinese Conference on Pattern Recognition and Computer Vision (PRCV), Springer, 2018, pp. 41–51.
[13] Y. Huang, M. He, L. Jin, Y. Wang, RD-GAN: few/zero-shot chinese character style transfer via radical decomposition and rendering, Computer Vision - ECCV 2020, 2020.
[14] X.-Y. Zhang, F. Yin, Y.-M. Zhang, C.-L. Liu, Y. Bengio, Drawing and recognizing chinese characters with recurrent neural network, IEEE Trans. Pattern Anal. Mach. Intell. 40 (4) (2017) 849–862.

[15] T. Isobe, X. Jia, S. Chen, J. He, Y. Shi, J. Liu, H. Lu, S. Wang, Multi-target domain adaptation with collaborative consistency learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 8187–8196.
[16] R. Luo, F. Tian, T. Qin, E. Chen, T.-Y. Liu, Neural architecture optimization, Adv. Neural Inf. Process. Syst. 31 (2018).
[17] T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A simple framework for contrastive learning of visual representations, in: International Conference on Machine Learning, PMLR, 2020, pp. 1597–1607.
[18] C. Doersch, A. Gupta, A.A. Efros, Unsupervised visual representation learning by context prediction, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1422–1430.
[19] R. Hadsell, S. Chopra, Y. LeCun, Dimensionality reduction by learning an invariant mapping, in: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), Vol. 2, IEEE, 2006, pp. 1735–1742.
[20] A. Aberdam, R. Litman, S. Tsiper, O. Anschel, R. Slossberg, S. Mazor, R. Manmatha, P. Perona, Sequence-to-sequence contrastive learning for text recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 15302–15312.
[21] A. Baevski, H. Zhou, A. Mohamed, M. Auli, wav2vec 2.0: A framework for self-supervised learning of speech representations (2020).
[22] D. He, Y. Xia, T. Qin, L. Wang, N. Yu, T.-Y. Liu, W.-Y. Ma, Dual learning for machine translation, Adv. Neural Inf. Process. Syst. 29 (2016) 820–828.
[23] J. Gehring, M. Auli, D. Grangier, D. Yarats, Y.N. Dauphin, Convolutional sequence to sequence learning, in: International Conference on Machine Learning, PMLR, 2017, pp. 1243–1252.
[24] A. Baevski, S. Schneider, M. Auli, vq-wav2vec: Self-supervised learning of discrete speech representations, arXiv preprint arXiv:1910.05453(2019a).
[25] A. Baevski, M. Auli, A. Mohamed, Effectiveness of self-supervised pre-training for speech recognition, arXiv preprint arXiv:1911.03912(2019b).
[26] C. Luo, L. Jin, Z. Sun, Moran: A multi-object rectified attention network for scene text recognition, Pattern Recognit. 90 (2019) 109–118.
[27] D. Tanaka, D. Ikami, T. Yamasaki, K. Aizawa, Joint optimization framework for learning with noisy labels, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5552–5560.
[28] M. Gutmann, A. Hyvärinen, Noise-contrastive estimation: a new estimation principle for unnormalized statistical models, in: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, 2010, pp. 297–304.
[29] T.-L. Yuan, Z. Zhu, K. Xu, C.-J. Li, T.-J. Mu, S.-M. Hu, A large chinese text dataset in the wild, J. Comput. Sci. Technol. 34 (3) (2019) 509–521.
[30] F. Yin, Q.-F. Wang, X.-Y. Zhang, C.-L. Liu, ICDAR 2013 Chinese handwriting recognition competition, in: 2013 12th International Conference on Document Analysis and Recognition, IEEE, 2013, pp. 1464–1470.
[31] C.-L. Liu, F. Yin, D.-H. Wang, Q.-F. Wang, CASIA online and offline chinese handwriting databases, in: 2011 International Conference on Document Analysis and Recognition, IEEE, 2011, pp. 37–41.
[32] W. Wang, J. Zhang, J. Du, Z.-R. Wang, Y. Zhu, DenseRAN for offline handwritten chinese character recognition, in: 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR), IEEE, 2018, pp. 104–109.
[33] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556(2014).
[34] A. Gupta, A. Vedaldi, A. Zisserman, Synthetic data for text localisation in natural images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2315–2324.
[35] L. Wu, C. Zhang, J. Liu, J. Han, J. Liu, E. Ding, X. Bai, Editing text in the wild, in: Proceedings of the 27th ACM International Conference on Multimedia, 2019, pp. 1500–1508.
[36] Y. Li, Y. Zhu, J. Du, C. Wu, J. Zhang, Radical counter network for robust chinese character recognition, in: 2020 25th International Conference on Pattern Recognition (ICPR), IEEE, 2021, pp. 4191–4197.
[37] C.-L. Liu, F. Yin, D.-H. Wang, Q.-F. Wang, Online and offline handwritten chinese character recognition: benchmarking on new databases, Pattern Recognit. 46 (1) (2013) 155–162.

Mobai Xue received his BEng degree from University of Science and Technology of China (USTC), in 2019. He is currently a Master's candidate at University of Science and Technology of China (USTC). His current research area includes deep learning and OCR.

Jun Du received his BEng and PhD degrees from the Department of Electronic Engineering and Information Science, University of Science and Technology of China (USTC) in 2004 and 2009, respectively. From 2004 to 2009, he was with iFlytek Speech Lab of USTC. During the above years, he worked as an Intern for two 9-month periods at Microsoft Research Asia (MSRA), Beijing. In 2007, he worked as a Research Assistant for 6 months in the Department of Computer Science, University of Hong Kong. From July 2009 to June 2010, he worked at iFLYTEK Research on speech recognition. From July 2010 to January 2013, he joined MSRA as an Associate Researcher, working on handwriting recognition, OCR, and speech recognition. Since February 2013, he has been with the National Engineering Research Center of Speech and Language Information Processing (NERC-SLIP) of USTC.