Reality Transform Adversarial Generators for Image Splicing Forgery Detection and Localization

Xiuli Bi  Zhipeng Zhang  Bin Xiao∗
Chongqing University of Posts and Telecommunications
Chongqing, China
bixl@cqupt.edu.cn  s190201087@stu.cqupt.edu.cn  xiaobin@cqupt.edu.cn

Abstract

When many forgery images become more and more realistic with help of image editing tools and convolutional neural networks (CNNs), authenticators need to improve their ability to verify these forgery images. The process of generating and detecting forgery images is the same as the principle of Generative Adversarial Networks (GANs). In this paper, since the retouching progress of forgery images requires to suppress the tampering artifacts and to keep the structural information, we consider this retouching progress as an image style transform, and then propose a fake-to-realistic transform generator $G_T$. For detecting the tampered regions, a localization generator $G_M$ is proposed too, which is based on a multi-decoder-single-task strategy. By adversarial training two generators, the proposed $\alpha$-learnable whitening and coloring transform ($\alpha$-learnable WCT) block in $G_T$ automatically suppress the tampering artifacts in the forgery images. Meanwhile, the detection and localization abilities of $G_M$ will be improved by learning the forgery images retouched by $G_T$. The experiment results demonstrate that the proposed two generators in GAN can simulate confrontation between the faker and the authenticator well; the localization generator $G_M$ outperforms the state-of-the-art methods in splicing forgery detection and localization on four public datasets.

1. Introduction

Cyberspace has experienced explosive growth, and countless images are uploaded to the Internet every day, which includes a lot of forgery images. Since forgery images can be easily produced by user-friendly image editing tools and used to create fake news and rumors, it is necessary to develop more effective methods for image forgery detection and localization. For the image forgeries, copy-move and removal forgery require a single source image, but splicing forgery copies and pastes regions from one or more source images onto the target image. Fig. 1-(a) demonstrates the two examples of splicing forgery images. In this paper, our work focuses on detecting the splicing forgery images and then locate the tampered regions of these detected images.

The image splicing forgery detection methods can be summarized into two main categories, methods based on conventional features extraction[19, 6, 14, 21] and methods based on convolutional neural networks (CNNs)[26, 28, 9, 1, 25, 24, 2, 13]. Most conventional methods focus on a particular image fingerprint that is caused by imaging processing and post-processing. Because the particular image fingerprint is easy to be influenced by post-processing, such as JPEG compression, down-sampling, and mean filtering, many conventional methods are easy to fail. Fig. 1-(c) shows the experiments results of a conventional method[19].

CNN-based methods can be further divided into patch-based methods and end-to-end methods. For patch-based methods, since the final detection result is derived from the decisions of image patches, the detected results are generally composed of square white blocks, or only the patches on boundaries of the tampered regions are detected. For end-to-end methods, if the tampering artifacts are suppressed and reduced by the faker, it is difficult for end-to-end methods to detect tampered regions. Fig. 1-(d) shows the experimental results of a CNN-based method[1].

To solve these problems, V. Kniaz et al.[13] introduced a GAN-based method named Mixed Adversarial Generators (MAG) for image splicing forgery detection and localization. However, MAG requires class segmentations to re-touch splicing forgery images, which consumes a host of computational resources. Furthermore, since the prediction of both tampered region and class segmentation is generated in a single decoder network, some untampered semantic regions, who are similar to the tampered regions in the ground truth, will be easily detected as the tampered regions, as the experiment results demonstrated in Fig. 1-(e)
In this work, we rethink the principle of generating and detecting forgery images. When image fakers retouch the forgery images more realistic, they need to hide the tampering artifacts, while keeping the structural information of the forgery image unchanged. The retouching progress of forgery images is the same as the task of image style transform. Thus, we consider the retouching process of forgery image as the image style transform, which transforms splicing forgery images from a ‘fake style’ to a ‘real style’. Based on this insight, we propose the fake-to-realistic transform generator $G_T$ to simulate the faker. In contrast, the authenticators need to detect the tampered regions from these more ‘real style’ splicing forgery images, so a localization generator $G_M$ with the multi-decoder-single-task (MDST) strategy is proposed. In the adversarial training between $G_T$ and $G_M$, for progressively suppressing the tampering artifacts of the splicing forgery image, we propose $\alpha$-learnable whitening and coloring transform blocks ($\alpha$0.-learnable WCT) based on WCT$^{[16]}$ in $G_T$. While, through the multi-decoder-single-task strategy (MDST), $G_M$ will improve its detection and localization ability by learning fewer tampering artifacts from the retouched images. Moreover, the discriminators $D_T$ and $D_M$ will qualify the outputs of $G_T$ and $G_M$. The GAN framework for adversarial training $G_M$ and $G_T$ is named Reality Transform Adversarial Generators (RTAG), the two examples of detection results are presented in Fig. 1-(f).

The main contribution of this work can be summarized as follows: (1) The retouching progress of forgery images is considered as the image style transform in this paper. Based on this insight, a fake-to-realistic transform generator $G_T$ is proposed, which applies the $\alpha$-learnable WCT blocks to automatically progressively retouch the splicing forgery images more realistic; (2) For detecting the tampered regions by fewer tampering artifacts, a localization generator $G_M$ is proposed according to the multi-decoder-single-task strategy; (3) By adversarial training $G_T$ and $G_M$ in the GAN framework, the localization generator $G_M$ will detect and locate the tampered regions even the splicing forgery images has fewer tampering artifacts.

2. Related Work

Most existing image splicing forgery detection methods can be divided into the methods based on conventional features extraction and the methods based on CNN. For conventional methods, Liu et al.$^{[20]}$ proposed aligned double quantization detection (ADQ), which utilizes the distribution of the image discrete cosine transform (DCT) coefficients to distinguish the tampered regions. Krawetz et al.$^{[14]}$ proposed an error level analysis method (ELA), which is intended to find the compression error difference between the forgery regions and the real regions. Cozzolino et al. $^{[4]}$ proposed a method for blind detection and localization of splicing that uses cooccurrence based features, it requires no prior knowledge of the host camera, the splicing, or their processing history.

For CNN-based methods, many CNN-based methods mainly learn the differences between image patches and then determine whether an image patch was manipulated or not. For instance, Bappy et al.$^{[1]}$ proposed a network that contains a long short-term memory network (LSTM) and an encoder-decoder architecture network. This proposed network exploits resampling features from image patches to detect tampered regions. Xiao et al.$^{[26]}$ proposed a two-stage detection network, which learns the differences of the image properties between un-tampered and tampered regions from image patches with different scales. To directly learn from the whole images and locate the tampered regions, some end-to-end splicing forgery detection methods are proposed. Wu et al.$^{[25]}$ presented ManTra-Net, which contains an image manipulation trace feature extraction network and a local anomaly detection network. Bi et al.$^{[2]}$ proposed a ringed residual structure U-Net (RRU-Net), which is an end-to-end image essence attribute segmentation network without any pre-processing and post-processing. The end-to-end methods can detect tampered regions by learning various tampering artifacts.
directly from the whole images. Hu et al.[9] proposed a spatial pyramid attention network (SPAN) architecture that compares patches through the local self-attention block on multiple scales.

GAN is a special framework of CNN, although recent researches[11, 15, 29, 22] have revealed that GANs can achieve amazing success in multiple tasks, GAN-based image splicing forgery detection is still rare. V. Kniaz et al.[13] introduced MAG for image splicing forgery detection and localization. MAG adversarial trains a retoucher to retouch the fake images and an annotator to predict the tampered regions. MAG requires class segmentations to reconstruct and retouch splicing forgery images, which consumes a host of computational resources and the quality of retouched images are not realistic enough.

3. Proposed Method

In the proposed RTAG framework, generating and detecting splicing forgery image is considered as an adversarial game between a fake-to-realistic transform generator $G_T$ and a localization generator $G_M$. $G_T$ progressively retouches splicing forgery images from a ‘fake style’ to a ‘real style’, then $G_M$ needs to detect the tampered regions by learning the images retouched by $G_T$, these retouched images have fewer tampering artifacts. By adversarial training of $G_T$ and $G_M$, the detection and localization abilities of $G_M$ will be enhanced. This RTAG framework is shown in Fig. 2. Here, $G_T$ and $G_M$ follow the objective function $V(G_M)$ and $V(G_T)$:

$$
\min_{G_M} V(G_M) = \frac{1}{3} \mathbb{E}_{x \sim X} [(G_M(x) - m)^2] \\
+ \frac{1}{3} \mathbb{E}_{x \sim X} [(G_M(G_T(x)) - m)^2] \\
+ \frac{1}{3} \mathbb{E}_{y \sim Y} [(G_M(y) - 0_{W,H})^2],
$$

(1)

$$
\min_{G_T} V(G_T) = \mathbb{E}_{x \sim X} [(G_M(G_T(x)) - 0_{W,H})^2].
$$

Where $x$ denotes the values in a splicing forgery image $I_x$; $y$ denotes the values of in an authentic image $I_y$; $X$ and $Y$ denote the forgery domain and the authentic domain separately; $m$ represents the ground truth of splicing forgery image $I_x$; $0_{W,H}$ is a black image that represents the authentic
image $I_y$ does not have any tampered regions.

### 3.1. Fake-to-realistic Transform Generator $G_T$

In MAG[13], the annotator generator makes sure the retouched images recognizable by generating the prediction of tampered regions and class segmentations. Generating class segmentations not only need additional computational sources but also may disturb the task of localization. Thus, in this paper, we consider the retouching progress as an image style transform. The fake-to-realistic transform generator $G_T$ is expected to transform splicing forgery images $I_x$ to realistic images while keeping the structural information of $I_x$ unchanged. As shown in Fig. 2-(a), we propose the fake-to-realistic transform generator $G_T$ that applies WCT[16] block between certain layers of the U-Net[23], and a global block is inserted between encoder and decoder. The structure of corresponding discriminator $D_T$ is a conditional discriminator same as PatchGAN[10] architecture.

In generator $G_T$, a splicing forgery image $I_x$ and an authentic image $I_y$ are randomly paired, and they input the first encoding block to generate feature map $f_x$ and $f_y$. WCT block directly matches feature map $f_x$ to the covariance matrix of feature map $f_y$. WCT firstly peels off the style features in $f_x$, such as colors, contrast, etc. Then the transform feature map $f_{xy}$ will be obtained by filling the peeled feature map $f_x$ with the style features in $f_y$. Finally, the transform feature map $f_{xy}$ is blended with feature map $f_x$ by Eq. (2).

$$
\hat{f}_{xy} = \alpha f_{xy} + (1 - \alpha) f_x
$$

Where $\hat{f}_{xy}$ denotes the output feature of the first WCT block. $\alpha \in [0, 1]$ denotes the weight that controls the degree of retouching. Then, $\hat{f}_{xy}$ will be the input feature $f_x$ of the next block.

The previous works[16, 17, 27] only manually set the value of $\alpha$. However, if $\alpha$ is too high, the structural information in the retouched images may be lost, the retouched images always carry black plaques and the edges of the retouched image are blurred with a color halo. Moreover, if $\alpha$ is near to 1, the features of the splicing forgery image $I_x$ are almost replaced by the authentic image, $G_M$ will learn nothing to distinguish the tampered regions. On the other hand, if $\alpha$ is too low, the WCT will lose its function. Therefore, it’s difficult to find a suitable value of $\alpha$ manually. To address this issue, we propose $\alpha$-learnable WCT block, the structure of this block is shown in Fig. 3. $\alpha$-learnable WCT block can determine the best value of $\alpha$ by learning the feature map $f_x$ and $f_y$. The qualitative result of $\alpha$-learnable WCT block is shown in Fig. 4-(f). Based on the experiment results for evaluating $\alpha$-learnable WCT block, we believe it can be further used in other end-to-end style transform networks.

Because the features of the forgery image $I_x$ should be replaced by the features of the authentic image $I_y$ from a global view. Based on the global block proposed in[3], we modify this global block by applying the convolutions of different receptive fields to extract multi-level features, the modified global block can get more comprehensive global features. It is inserted between encoder and decoder to extract global features to enhance the realistic transform efficiency. The structure of the modified global block is shown in Fig. 5.

For implementing multi-task in $G_T$, $G_T$ uses a mixed loss function, which consists of four parts: $L_{\text{content}}$, $L_{\text{recons}}$, $L_{\text{realistic}}$ and $L_{\text{adv}}$. $G_T$ should not change the
structural information of the forgery images while retouching the forgery images, so the content loss function $L_{content}$ is defined as Eq. (3).

$$L_{content}^{GT} = E_{x \sim X} [\| x - G_T(x) \|_1]$$

(3)

In Eq. (3), $x$ denotes the values of a splicing forgery image $I_x$. $\| \cdot \|_1$ denotes the $L_1$ norm. Since $G_T$ needs to reconstruct the authentic image $I_y$, and keep that the reconstructed image $I_y^G$ is the same as the authentic image $I_y$. So, the loss function of reconstruction $L_{recons}$ is applied to reinforce the reconstruction ability of $G_T$. $L_{recons}^{GT}$ is defined in Eq. (4).

$$L_{recons}^{GT} = E_{y \sim \hat{Y}} [\| y - G_T(y) \|_1]$$

(4)

Where $y$ denotes the values of the authentic image $I_y$. Since $G_T$ is adversarial trained against $G_M$, when the output retouched image $I_x^G$ is more realistic, the prediction of $G_M$ is harder. Therefore, we use a realistic loss function $L_{realistic}^{GT}$ conducted by Eq. (5)

$$L_{realistic}^{GT} = E_{x \sim X} [\| W_H - G_M(G_T(x)) \|_1]$$

(5)

Finally, we use the least-squares equation, which is defined in Eq. (6), as adversarial loss function of corresponding discriminator $D_T$. The adversarial loss function $L_{adv}$ will make the output retouched image $I_x^G$ more realistic.

$$L_{adv}^{GT} = \frac{1}{2} E_{x \sim X} [(D_T(G_T(x)) - 1)^2].$$

(6)

The final loss function can be summarized as:

$$L_T = \lambda_{content} L_{content}^{GT} + \lambda_{recons} L_{recons}^{GT} + \lambda_{realistic} L_{realistic}^{GT} + \lambda_{adv} L_{adv}^{GT},$$

(7)

$$\lambda_{content} = 1, \lambda_{recons} = 0.5, \lambda_{realistic} = 0.5, \lambda_{adv} = 1.$$
loss. 

\[
L_{edge}^{GM} = -\frac{1}{2W \times H} \mathbb{E}_{x \sim X} [\omega_{pos} \cdot GT_{edge} \cdot \log(G_M(x)) + \omega_{neg} \cdot (1 - GT_{edge}) \cdot \log(1 - G_M(x)) + \omega_{pos} \cdot GT_{edge} \cdot \log(G_M(G_T(x))) + \omega_{neg} \cdot (1 - GT_{edge}) \cdot \log(1 - G_M(G_T(x)))]
\]

(9)

Where, \( G_M(x) \) is the detected edges of the tampered regions in forgery image \( I_x \). \( G_M(G_T(x)) \) is the detected edges of the tampered regions in the retouched image \( I_x^R \). \( GT_{edge} \) denotes the edges of tampered regions in the ground truth. \( \omega_{pos} \) and \( \omega_{neg} \) are the weights that make \( G_M \) focus more on the edges of the tampered regions. In the experiments below, we set \( \omega_{pos} = 1.5 \) and \( \omega_{neg} = 0.5 \). For reconstruction decoder in \( G_M \), the loss function \( L^{G_M}_{recons} \) is calculated by Eq. (10).

\[
L^{G_M}_{recons} = \frac{1}{2} \mathbb{E}_{x \sim X} [\|x - G_M(x)\|_1 + \|G_T(x) - G_M(G_T(x))\|_1]
\]

(10)

In Eq. (10), \( G_M(x) \) denotes the reconstructed image \( \tilde{I}_x \). \( G_M(G_T(x)) \) is the reconstructed image \( \tilde{I}_x^R \). Finally, an adversarial loss \( L^{G_M}_{adv} \) is proposed to avoid blurry outputs, which is defined as Eq. (11).

\[
L^{G_M}_{adv} = \frac{1}{2}(D_M(C_5) - 1)^2,
\]

(11)

Where, \( C_5 \) is a concatenation input of three parts: the detected regions of forgery image \( I_x \), the detected edges of the tampered regions in forgery image \( I_x \), the splicing forgery image \( I_x \) or the retouched image \( I_x^R \). The subscript of \( C_5 \) is the channel number of the concatenation. Finally, The final loss function of \( G_M \) is computed as follow:

\[
L_M = \lambda_{mask}^{G_M} L_{mask}^{G_M} + \lambda_{edge}^{G_M} L_{edge}^{G_M} + \lambda_{recons}^{G_M} L_{recons}^{G_M} + \lambda_{adv}^{G_M} L_{adv}^{G_M}
\]

(12)

\( \lambda_{mask}^{G_M} = 1, \lambda_{edge}^{G_M} = 1, \lambda_{recons}^{G_M} = 0.1, \lambda_{adv}^{G_M} = 0.1 \). Each weight \( \lambda \) in \( L_M \) is set by the experience of the experiments.

4. Experiments
4.1. Datasets

For the fair comparison, we perform evaluations on four public splicing forgery image datasets: CASIA v2.0[5], Columbia[8], NIST 2016[7] and FantasticReality[13]. The details of each dataset are illustrated in Table 1. CASIA v2.0 contains three types of image forgeries: splicing, copy-move, and removal, the forgery images are post-processed by methods such as filtering and blurring. The Columbia dataset only contains splicing forgery and the tampered regions are the large meaningless smooth regions which is not post-processed. The image forgery types of NIST 2016 include splicing, copy-move, and removal, all the forgery images in the dataset are post-processed to hide visible traces of manipulation. FantasticReality contains a large number of forgery images but only splicing forgery is included, the splicing forgery images are not post-processed by any method. Because we aim to detect splicing forgery, only the splicing forgery images are selected in each dataset.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Image Format</th>
<th>Forgery/Authentic Images</th>
<th>Train/Test Images</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASIA v2.0</td>
<td>TIFF, JPEG</td>
<td>5123/7491</td>
<td>715/100</td>
</tr>
<tr>
<td>Columbia</td>
<td>JPEG</td>
<td>180/183</td>
<td>125/45</td>
</tr>
<tr>
<td>NIST 2016</td>
<td>JPEG</td>
<td>564/875</td>
<td>184/50</td>
</tr>
<tr>
<td>FantasticReality</td>
<td>JPEG</td>
<td>19422/16592</td>
<td>12000/1000</td>
</tr>
</tbody>
</table>

Table 1. Characteristics of the image splicing forgery datasets.

4.2. Evaluation Metrics

The performance of splicing localization is evaluated by mean average precision(mAP), Area Under Curve(AUC), and \( F \) rate defined by the following equations:

\[
Precison = \frac{TP}{TP + FP}, \\
Recall = \frac{TP}{TP + FN}, \\
F = \frac{2 \times Precison \times Recall}{Precison + Recall}
\]

(13)

where, \( TP \) and \( FP \) denote the number of correctly detected and erroneously detected pixels, and \( FN \) is the number of falsely missed pixels.

4.3. Setup

In our experiments, RTAG is trained using an Adam[12] training optimizer with a batch size of 8, an initial learning rate of 3e-4, a decay rate of 0, and an epoch of 300. Note that, in our observation, the performance of \( G_T \) drops sharply when batch size is more than 1, and \( G_M \)'s performance drops sharply when batch size is too low. Therefore, \( G_T \) is trained with one splicing forgery image and one authentic image each time, while \( G_M \) is trained with 8 splicing forgery images or 8 retouched images each time. To avoid \( G_T \) takes excessive advantage in the adversarial training, \( G_T \) is updated every 8 batches. For data augmentation, all the images are resized to 512×512. All training processes are implemented on a NVIDIA Tesla V100 (32G) GPU.

4.4. Comparisons

We compare RTAG with four state-of-the-art deep learning splicing forgery detection methods: ManTra[25],
Table 2. Experimental results of plain splicing forgery.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>CASIA v2.0</th>
<th>Columbia</th>
<th>NIST</th>
<th>FantasticReality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metric</td>
<td>mAP</td>
<td>AUC</td>
<td>F</td>
<td>mAP</td>
</tr>
<tr>
<td>ADQ</td>
<td>0.293</td>
<td>0.698</td>
<td>0.476</td>
<td>0.344</td>
</tr>
<tr>
<td>ELA</td>
<td>0.054</td>
<td>0.306</td>
<td>0.158</td>
<td>0.302</td>
</tr>
<tr>
<td>ManTra</td>
<td>0.569</td>
<td>0.777</td>
<td>0.651</td>
<td>0.468</td>
</tr>
<tr>
<td>LSTM</td>
<td>0.526</td>
<td>0.758</td>
<td>0.617</td>
<td>0.488</td>
</tr>
<tr>
<td>C2RNet</td>
<td>0.572</td>
<td>0.793</td>
<td>0.676</td>
<td>0.507</td>
</tr>
<tr>
<td>MAG</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>RTAG</td>
<td>0.707</td>
<td>0.888</td>
<td>0.815</td>
<td>0.796</td>
</tr>
</tbody>
</table>

MAG[13], LSTM[1], C2RNet[26], and two conventional methods: ADQ[19] and ELA[14]. Moreover, we especially compare our method with MAG on the FantasticReality dataset only, because MAG needs class segmentations which are only provided in the FantasticReality dataset. All the methods we compared are implemented with the code and parameters proposed in the original papers.

We evaluate the performance of RTAG and comparative methods at pixel-level. The evaluation results are present in Table 2. The conventional methods always detect the whole image as a tampered region, so these methods contain very high Recall but low Precision. The training set of NIST 2016 is very small, and the tampered images are proper post-processed to hide tampering artifacts, so many methods fail on this dataset. But our model learns to detect tampered regions by fewer tampering artifacts and outperforms other methods on NIST 2016. The results shown in Fig. 6 and Fig. 7 indicate that the performance of our method is better than the state-of-the-art methods.
4.5. Ablation Study

To evaluate the necessity of each component of RTAG, we compare the splicing forgery detection performance of several ablated versions of RTAG on CASIA v2.0. The detection results are presented in Table 3 and Fig. 8. We first evaluate the performance of $G_M$ that is only trained by splicing forgery images without $G_T$. The result demonstrates that the adversarial training between $G_M$ and $G_T$ is critical for RTAG. Then we evaluate $G_M$ in SDMT strategy, which means all the outputs of $G_M$ are generated by a single decoder. The result proves that the MDST strategy significantly improves the performance of the model.

<table>
<thead>
<tr>
<th>Method</th>
<th>Components</th>
<th>Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$G_T$</td>
<td>$G_M$</td>
</tr>
<tr>
<td></td>
<td>Edge</td>
<td>Recons</td>
</tr>
<tr>
<td>Without $G_T$</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SDMT</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Without Recons</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Without Edge</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>RTAG</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 3. Evaluation results for ablated versions of RTAG.

5. Conclusion

In this paper, we present a novel generative adversarial network framework ((RTAG)) for splicing detection and localization. RTAG adversarial trains a fake-to-realistic translation generator $G_T$ and a localization generator $G_M$ to simulate the image fakers and the image authenticators. A novel $\alpha$-learnable whitening and coloring transform block is proposed in $G_T$ to automatically and progressively suppress the tampering artifacts of the forgery images. Meanwhile, the multi-decoder-single-task strategy of $G_M$ will push $G_M$ to improve its detection and localization abilities by learning the retouched images with less tampering artifacts. The multi-decoder-single-task strategy, and $G_M$ can learn to detect tampered regions from fewer tampering artifacts by adversarial training against $G_T$. Experimental results demonstrate that the proposed method outperforms the state-of-the-art methods on image splicing forgery detection and localization.

Acknowledgements

This work was supported in part by the National Key Research and Development Project under Grant 2019YFE0110800 and Grant 2016YFC1000307-3, in part by the National Natural Science Foundation of China under Grant 61806032 and Grant 61976031, in part by the National Major Scientific Research Instrument Development Project of China under Grant 62027827, in part by the Scientific and Technological Key Research Program of Chongqing Municipal Education Commission under Grant KJZD-K201800601. Our deepest gratitude goes to the anonymous reviewers for their careful work and thoughtful suggestions that have helped improve this paper substantially.
References


