Robust Image Forgery Detection Against Transmission Over Online Social Networks

Haiwei Wu, *Student Member, IEEE*, Jiantao Zhou, *Senior Member, IEEE*, Jinyu Tian, *Student Member, IEEE*, Jun Liu, *Student Member, IEEE*, and Yu Qiao, *Senior Member, IEEE*

***Abstract*— The increasing abuse of image editing software causes the authenticity of digital images questionable. Meanwhile, the widespread availability of online social networks (OSNs) makes them the dominant channels for transmitting forged images to report fake news, propagate rumors, etc. Unfortunately, various lossy operations, e.g., compression and resizing, adopted by OSNs impose great challenges for implementing the robust image forgery detection. To fight against the OSN-shared forgeries, in this work, a novel robust training scheme is proposed. Firstly, we design a baseline detector, which won the top ranking in a recent certificate forgery detection competition. Then we conduct a thorough analysis of the noise introduced by OSNs, and decouple it into two parts, i.e., *predictable noise* and *unseen noise*, which are modelled separately. The former simulates the noise introduced by the disclosed (known) operations of OSNs, while the latter is designed to not only complete the previous one, but also take into account the defects of the detector itself. We further incorporate the modelled noise into a robust training framework, significantly improving the robustness of the image forgery detector. Extensive experimental results are presented to validate the superiority of the proposed scheme compared with several state-of-the-art competitors, especially in the scenarios of detecting OSN-transmitted forgeries. Finally, to promote the future development of the image forgery detection, we build a public forgeries dataset based on four existing datasets through the uploading and downloading of four most popular OSNs. The data and code of this work are available at https://github.com/HighwayWu/ImageForensicsOSN.**

***Index Terms*— Image forgery detection, social networks, deep neural networks, robustness.**

I. Introduction

THE ever-increasing popularity of powerful image editing software, such as Photoshop and Meitu, has made the manipulation of images an extremely easy task. The manipulated or forged images are becoming increasingly dangerous in various fields such as removing copyright watermarks, producing fake news, and being forged evidence in court; negatively affecting not only individuals but also the whole society. Meanwhile, with the vigorous development of the Internet, online social networks (OSNs) have become dominant platforms for information transmission, where images occupy a large portion. Naturally, many forged images are transmitted over various OSNs, seriously influencing people's opinion towards e.g., important documents (certificates), commercial products, political issues, etc.

A large number of methods [1]–[16] have been proposed to detect and localize image forgery, so as to ensure information authenticity. Some of these forensic techniques are designed to detect specific forms of tampering, such as splicing [2], [6], copy-move [3], [7] and inpainting [5], [9], [10], while the others are to identify more complex or compound forgeries. However, few research has been done to explicitly address the design of robust forgery detection against the lossy operations in the ubiquitous OSN platforms. Such a topic is very important because these lossy operations can severely degrade the detection performance. As shown in Fig. 1, the state-of-the-art algorithm [1] can accurately detect the forged regions from the original forgery, but the detection performance would be severely degraded when handling the forgery transmitted through Facebook.

For mitigating the negative impacts of OSNs, the most critical issue is to analyze and model the noise introduced by the OSN lossy channels. However, this is a rather difficult problem mainly because the current platforms do not disclose the process for manipulating the transmitted images. Although some existing works [17], [18] revealed part of the processes adopted by OSNs, there are still many unknown operations, e.g., for Facebook, the enhancement filtering, the allocation mechanism of quality level, the resizing factor, and even the interpolation used in resizing, are all unclear. More importantly, OSNs often adjust their image processing pipelines, making the modeling even more challenging.

To deal with the aforementioned challenges, in this paper, we aim to design a robust image forgery detection method to defeat the lossy operations in OSNs. We first design a baseline detector, which won the top ranking in a recent certificate forgery detection competition [19]. Then for dealing with the OSN degradations, we propose a noise modeling scheme and integrate the mimetic noises into a robust training

Manuscript received September 28, 2021; revised December 17, 2021; accepted January 11, 2022. Date of publication January 19, 2022; date of current version February 8, 2022. This work was supported in part by the Macau Science and Technology Development Fund under Grant SKL-IOTSC-2021-2023, Grant 0072/2020/AMJ, Grant 077/2018/A2, and Grant 0060/2019/A1; in part by the Research Committee at the University of Macau under Grant MYRG2018-00029-FST and Grant MYRG2019-00023-FST; in part by the Natural Science Foundation of China under Grant 61971476; and in part by Alibaba Group through Alibaba Innovative Research Program. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Alessandro Piva. *(Corresponding author: Jiantao Zhou.)*

Haiwei Wu, Jiantao Zhou, Jinyu Tian, and Jun Liu are with the State Key Laboratory of Internet of Things for Smart City and the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Macau, China (e-mail: yc07912@umac.mo; jtzhou@umac.mo; yb77405@umac.mo; yc07453@umac.mo).

Yu Qiao is with the Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518000, China (e-mail: yu.qiao@siat.ac.cn).

Digital Object Identifier 10.1109/TIFS.2022.3144878

framework. More specifically, we decouple the OSN noises into two components: 1) *predictable noise* and 2) *unseen noise*. The former is designed to simulate the predictable loss brought by known operations, (e.g., JPEG compression and scaling), whose modeling relies on a deep neural network (DNN) with the residual learning and an embedded differentiable JPEG layer. While the latter is a supplement and extension of the former, mainly in response to the unknowable actions conducted by OSNs and/or the discrepancy in the training and testing of various OSNs. Apparently, it is unrealistic to build a suitable model for the unseen noise from the perspective of signal characteristics. To address this difficulty, we transfer our observations from the noise perspective to the detector itself, only focusing on the noise that may cause deterioration of the detection performance. Such a strategy naturally incubates a new algorithm to model the unseen noise by utilizing the core idea of *adversarial noise* [20], which is essentially imperceptible perturbation that can severely degrade the model performance.

As expected and will be verified by experiments, our robust image forgery detection method demonstrates superior robustness and outperforms several state-of-the-art algorithms, especially in the case of OSN transmission. An example of the detection result of our scheme is shown in Fig. 1, which validates the robustness of our model against the transmission over OSN. Finally, for further research in this area, we build a public forgeries dataset with more than 7000 items based on four existing datasets [21]–[24], through manually uploading and downloading over the platforms of Facebook, Whatsapp, Weibo, and Wechat, respectively.

Our major contributions can be summarized as follows:

- We design a baseline image forgery detector, which won the top ranking in a recent certificate forgery detection competition. This baseline detector also serves as the cornerstone of this work.
- We propose a novel training scheme for robust image forgery detection against transmission over OSNs. The training scheme not only models the predictable noise involved by OSNs, but also incorporates the unseen noise through a newly proposed algorithm to further promote the robustness of the detector.
- Our proposed model achieves better detection performance in comparison with several state-of-the-art methods [1], [14]–[16], especially in the scenario of fighting against the transmission over OSNs.
- We build a public forgery dataset based on four existing datasets [21]–[24], through uploading and downloading over the platforms of Facebook, Whatsapp, Weibo, and Wechat, respectively.

The rest of this paper is organized as follows. Section II reviews the related works on the image forgery detection and the manipulations of OSN. Section III presents the architecture of the baseline detector and Section IV details the proposed robust training scheme via the noise modeling. Experimental results are given in Section V and Section VI concludes.

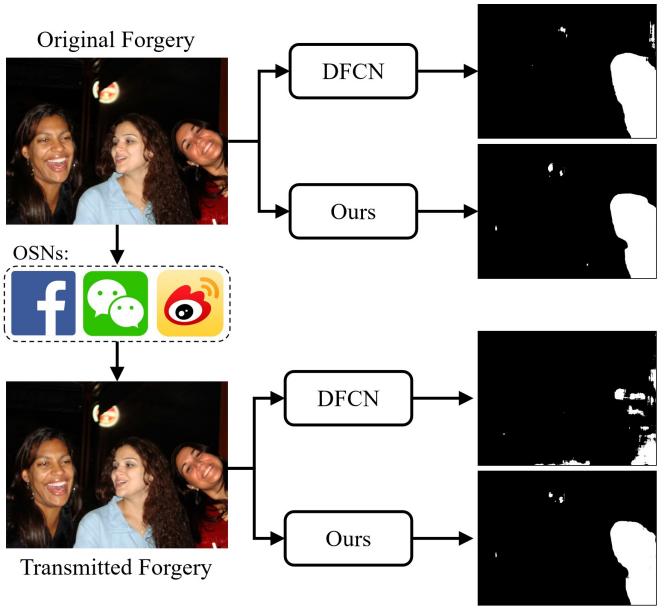

Fig. 1. The detection results of DFCN [1] and ours by using an original forgery and the forgery transmitted through OSN. The right woman in the forgery is spliced (forged).

II. Related Works

A. Image Forgery Detection

Many forensic methods (e.g., [2]–[10] and references therein) have been proposed to verify the authenticity of digital images. These methods detect the forged regions through the *specific* artifacts left by the tampering operations, e.g., splicing [2], [6], copy-move [3], [7], median filtering [4], [8], inpainting [5], [9], [10], etc. More specifically, Lyu *et al.* [2] introduced an effective method for the splicing detection by revealing inconsistencies in local noise levels. Through solving the keypoint matching problems over a massive number of keypoints, Li and Zhou [3] developed a fast hierarchical matching strategy for the detection of copy-move forgeries. As for the forensic detection of the median filtering, Kang *et al.* [4] adopted an autoregressive model to analyze the statistical properties of the median filter residual. To extract the evidence of the inpainting forgeries, Li *et al.* [5] proposed a diffusion-based detection method by analyzing the local variance of the image Laplacian along the isophote direction. With the success of neural networks in various fields, many deep learning based approaches [6]–[10] have been developed for detecting these specific forgeries. Unfortunately, these forensic approaches can only be applied to detect specific tampering manipulations, severely limiting their practical usefulness, as the prior knowledge regarding the forgery types is usually unavailable.

To better fit the practical requirements, in recent years, more and more methods have been developed to address the problem of detecting general (compound) types of forgeries [1], [11]–[16], among which the deep learning based methods are the most successful. Along this line of research, Wu *et al.* [14] proposed the MT-Net, a general forgery detection/localization network, which first extracts image manipulation features and then identifies anomalous regions by

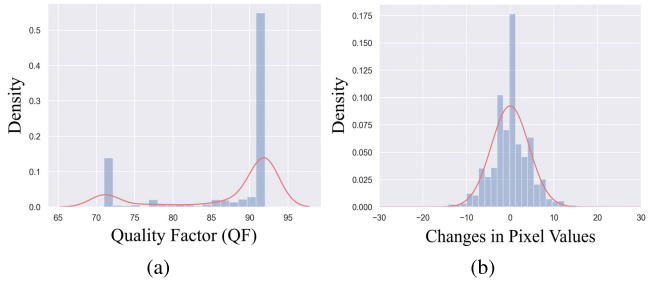

Fig. 2. The distribution of QF (a) and the changes in pixel values (b).

assessing how different a local feature is from its reference features. Mayer and Stamm recently [15] introduced the forensic similarity to determine whether two image patches contain the same or different forensic traces. From the perspective of the camera fingerprint, Cozzolino and Verdoliva designed a method for extracting a camera model fingerprint, called noiseprint, so as to disclose the forged regions via suppressing the scene contents while enhancing the model-related artifacts [16]. For learning the traces of generic forgeries, Zhuang *et. al* [1] utilized a training data generation strategy by resorting to Photoshop scripting.

B. Online Social Network (OSN)

The popularity of various OSN platforms, e.g., Facebook, Whatsapp, Wechat, Weibo, etc, significantly simplifies the dissemination and sharing of images. However, as indicated by many existing works [17], [18], almost all OSNs manipulate the uploaded images in a lossy fashion. The noise introduced by these lossy operations could severely affect the effectiveness of forensic methods. Taking Facebook as an example, as discovered in our previous works [17], [18], [25], these manipulations mainly consist of four stages: format conversion, resizing, enhancement filtering, and JPEG compression. Specifically, the uploaded image is first converted into the pixel domain, where the truncation is used to ensure the pixel values are within $[0, 255]$. After that, resizing would be applied if the resolution of the image is above 2048 pixels. Subsequently, some selected blocks in the image undergo highly adaptive and complex enhancement filtering. As mentioned in [17], [18], it is very challenging to precisely know these enhancement filtering operations due to their adaptiveness. Finally, the image is subject to a round of JPEG compression with a quality factor (QF) *adaptively* determined according to the image content. Through the analysis of the dataset provided in [18], the QF values used by Facebook range from 71 to 95, where a more detailed distribution is shown in Fig. 2(a). Furthermore, we also present in Fig. 2 (b) how the pixel values change when an image is transmitted through Facebook. For more details on OSN manipulations, please refer to [17], [18].

Although the image manipulations on different OSN platforms are different, the operations conducted by mainstream OSNs still share many similarities (e.g., ubiquitous JPEG compression) [18].

III. Baseline Image Forgery Detector

Before diving into the robust design of the image forgery detection, we first present the details of the baseline

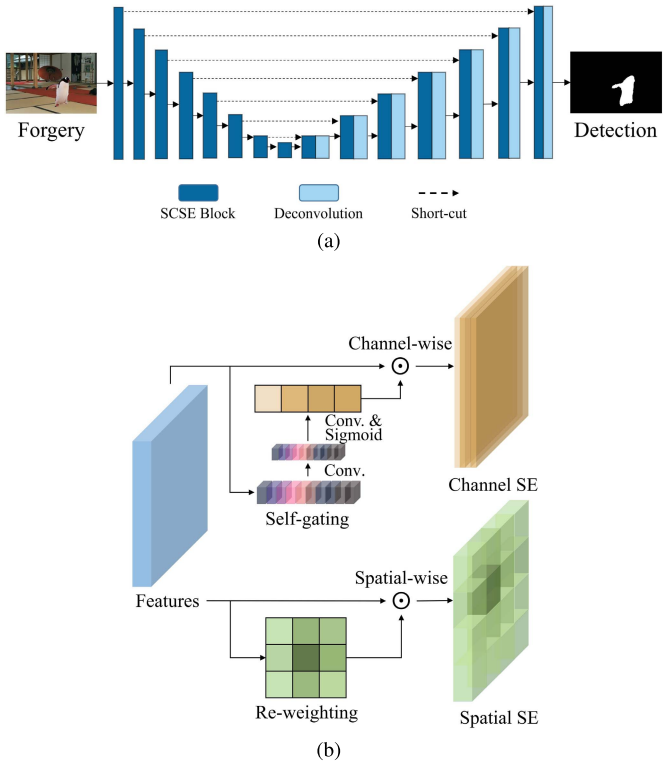

Fig. 3. The architecture of the detector f_{θ} (a) and the illustration of the SCSE block (b).

detector, which is the basis of the whole scheme. The detection network aims to detect the forged regions at the pixel level accuracy. The schematic diagram of the baseline image forgery detector is illustrated in Fig. 3. Specifically, the detector f_{θ} : $\mathbb{R}^{H\times W\times 3} \rightarrow \mathbb{R}^{H\times W\times 1}$ takes a color image with the resolution $H \times W$ as input, and eventually outputs the binary map for the detection result.

Since the forgery detection is essentially a two-class image segmentation problem, in our designed detector, we adopt the U-Net [26], one of the most commonly used structures for image segmentation, as the backbone architecture. U-Net consists of four consecutive encoders and four symmetric decoders, where each encoder contains repeated convolutional layers, the ReLU activation [27] and the max pooling operation. At the encoding stage, the spatial dimensions are constantly reduced for extracting more important feature information. At the decoding stage, by re-invoking the learned features from the corresponding encoder as the extra contextual information, the decoder can better optimize the results in various tasks. It should be noted that the input and output layers of the adopted U-Net backbone still need to be further optimized, so as to obtain satisfactory detection performance.

It was pointed out that the standard convolutional layer normally learns the features for representing the contents of input images, instead of the underlying forgery traces [13]. To improve the capability of extracting forgery relevant features, we further augment the architecture by incorporating the "Spatial Channel Squeeze-and-Excitation" (SCSE) mechanism [28], rather than simply using the traditional vanilla U-Net. The resulting variant U-Net called *SE-U-Net*, as shown

in Fig. 3, could selectively emphasize the informative features while suppressing the rest.

Specifically, the utilized SCSE layer [28] is composed of two branches, each of which performs the feature recalibration in the spatial and channel domains, respectively. For a given latent feature map $\mathbf{F} \in \mathbb{R}^{H\times W\times C}$, the spatial recalibration module first generates a re-weighting matrix $\mathbf{S} \in \mathbb{R}^{H\times W}$ by

$$\mathbf{S} = \text{Sigmoid}(\mathbf{W}_1 \otimes \mathbf{F}), \tag{1}$$

where $\mathbf{W}_1$ refers to the weights of a convolutional layer and $\otimes$ denotes the convolutional operator. Then the re-weighting matrix $\mathbf{S}$ is multiplied by the feature map $\mathbf{F}$ in a spatial-wise manner, to realize an adaptive excitation, and the resulting recalibrated spatial features are denoted by $\mathbf{F}_S$. That is,

$$\mathbf{F}_S = \text{Sigmoid}(\mathbf{W}_1 \otimes \mathbf{F}) \odot_s \mathbf{F}, \tag{2}$$

where $\odot_s$ means the spatial-wise multiplication.

On the other hand, by introducing a global average pooling layer, the channel recalibration first produces an intermediate vector $\mathbf{v} \in \mathbb{R}^{1\times 1\times C}$. The vector $\mathbf{v}$ is further refined by using a self-gating operation based on the channel dependence, namely,

$$\mathbf{v}^* = \text{Sigmoid}(\mathbf{W}_2 \otimes \text{ReLU}(\mathbf{W}_3 \otimes \mathbf{v})), \tag{3}$$

where $\mathbf{W}_2$ and $\mathbf{W}_3$ denote the weights of two fully connected layers. Eventually, the channel recalibrated features $\mathbf{F}_C$ is obtained by the channel-wise multiplication between $\mathbf{F}$ and $\mathbf{v}^*$. Specifically,

$$\mathbf{F}_C = \mathbf{v}^* \odot_c \mathbf{F}, \tag{4}$$

where $\odot_c$ means the channel-wise multiplication.

Thanks to the well-designed architecture for the image forgery detection, our baseline forgery detector trained appropriately won third place in the "Forgery Detection on Certificate Image" competition among 1561 teams around the world. This competition jointly organized by Alibaba Group and Tsinghua University is the first one for detecting and localizing forged regions in credentials and qualification documents (see the link [19] for details). In fact, the first place and the second place awarding schemes both somehow utilized the original, untouched images to assist the forgery detection and localization, which is not realistic in practical scenarios. In other words, excluding the original image-assisted schemes, our proposed baseline forgery detector is arguably the best one among all competing algorithms.

It should also be emphasized that although the baseline detector achieves good forgery detection and localization performance, it may not be robust enough against distortions, e.g., the ones incurred by OSN transmissions.

IV. Robust Image Forgery Detection Against Transmission Over OSNs

In this section, we tackle the challenging problem of designing a robust image forgery detection scheme against the transmission over various OSNs. The key technique leading to the success is to appropriately model the degradations incurred by OSNs, and integrate such knowledge into a robust training framework. As will be clear soon, such a robust training scheme can significantly improve the robustness of our baseline image forgery detector against the lossy operations over various OSNs.

More specifically, to enable an effective robust training strategy, we should appropriately model the noise incurred by OSN platforms. Recall from Section II-B that the image processing operations in an OSN are rather complicated; some of them can be precisely known, while some others can only be partially known or even completely unknown. Therefore, we propose to divide the OSN noise into two types: 1) predictable noise and 2) unseen noise. The former type corresponds to the case that the degradation source is clearly identified. While the latter type is a combination of various noise uncertainties caused by many factors, including the unknown modeling/parameters, the discrepancy between the training OSN and the testing OSN, and even some totally unseen degradation sources. By adding the modelled OSN noise in the training phase, the detector can learn more generalized features that survive the OSN transmission, making the overall forgery detection performance significantly improved.

Formally, let $\boldsymbol{\tau}$ and $\boldsymbol{\xi}$ denote the predictable noise and unseen noise, respectively, and hence the compound noise considered in the robust training stage becomes

$$\boldsymbol{\delta} = \boldsymbol{\tau} + \boldsymbol{\xi}. \tag{5}$$

For each training iteration, we first sample two pristine 3-channel (RGB) color images $\{\mathbf{p}_1, \mathbf{p}_2\} \in \mathbb{R}^{H\times W\times 3}$, and one binary mask $\mathbf{y} \in \{0,1\}^{H\times W\times 1}$, where 1's are assigned to the forged regions and 0's elsewhere. It should be noted that the forged regions could be spatially unconnected. Then a forged image $\mathbf{x}$ can be synthesized as

$$\mathbf{x} = \mathbf{p}_1 \odot (1 - \mathbf{y}) + \mathbf{p}_2 \odot \mathbf{y}, \tag{6}$$

where $\odot$ denotes the element-wise multiplication. Upon having pairs of forged image and the corresponding ground-truth mask, we can create a dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$ for the training, where i is the index for the training sample. Hence, the robust training of the image forgery detector $f_{\boldsymbol{\theta}}$ under the compound noise $\boldsymbol{\delta}$ can be formulated as:

$$\arg\min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{P(\boldsymbol{\delta})}\Big\{\mathcal{L}_b(f_{\boldsymbol{\theta}}(\mathbf{x}_i + \boldsymbol{\delta}), \mathbf{y}_i)\Big\}, \tag{7}$$

where $P(\boldsymbol{\delta})$ denotes the distribution of the compound noise $\boldsymbol{\delta}$, N is the number of training samples, and $\mathcal{L}_b$ is the binary cross-entropy (BCE) loss.

In our noise model, we consider a rather general setting that the two noise components $\boldsymbol{\tau}$ and $\boldsymbol{\xi}$ are dependent. Then, our robust training scheme (7) can be further written as

$$\arg\min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{P(\boldsymbol{\tau})}\Big\{\mathbb{E}_{P(\boldsymbol{\xi}|\boldsymbol{\tau})}\{\mathcal{L}_b(f_{\boldsymbol{\theta}}(\mathbf{x}_i + \boldsymbol{\tau} + \boldsymbol{\xi}), \mathbf{y}_i)\}\Big\}, \tag{8}$$

where $P(\boldsymbol{\tau})$ is the marginal distribution of $\boldsymbol{\tau}$, and $P(\boldsymbol{\xi}|\boldsymbol{\tau})$ is the conditional distribution of $\boldsymbol{\xi}$ given $\boldsymbol{\tau}$. As will be clear soon, from the implementation perspective, such expected values could be efficiently and accurately computed upon having an enough number of noise samples.

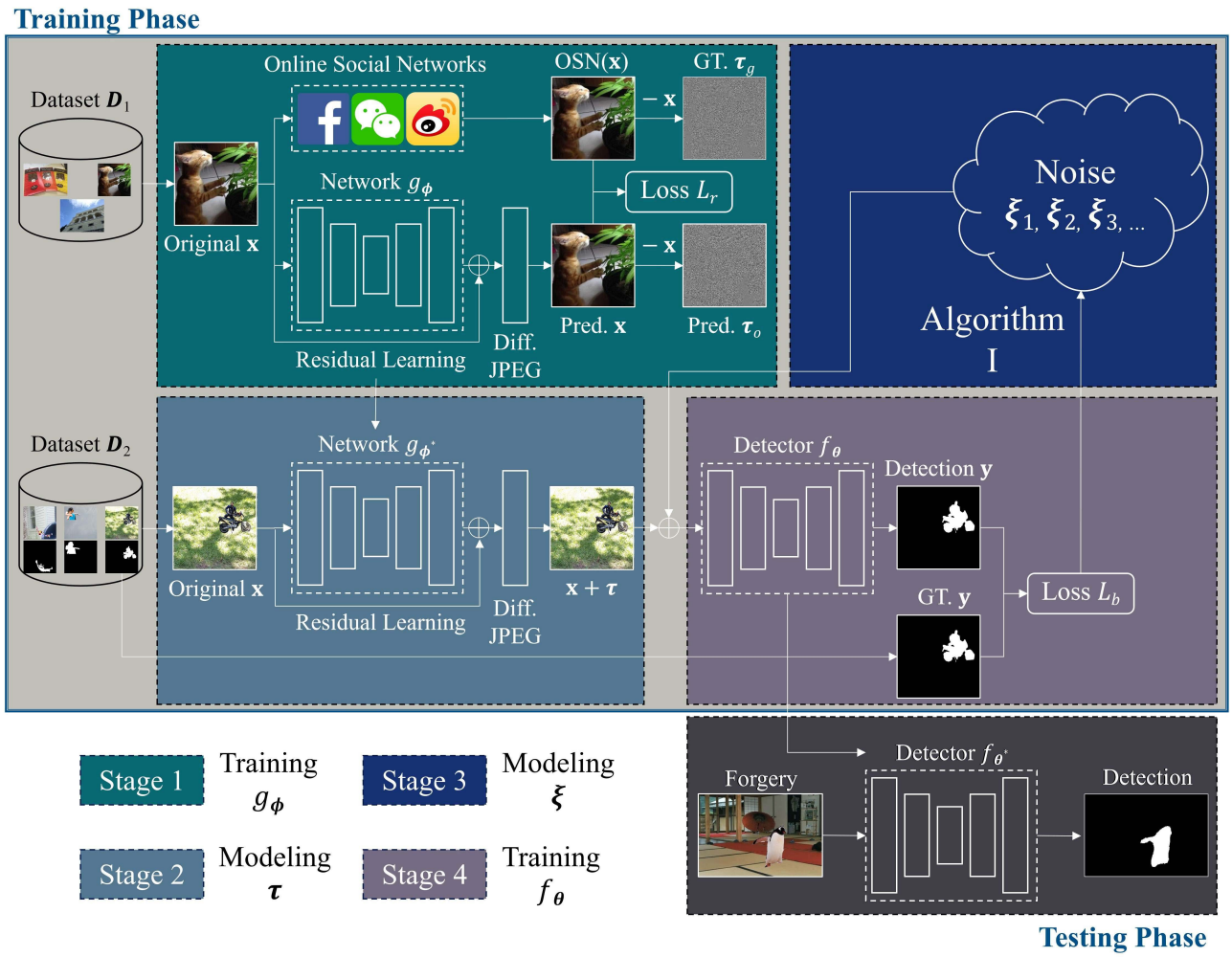

Fig. 4. The overview of our proposed training scheme and the corresponding testing phase.

To conduct the robust training given in (8), a crucial task is to model the marginal distribution $P(\boldsymbol{\tau})$ and the conditional distribution $P(\boldsymbol{\xi}|\boldsymbol{\tau})$, or equivalently (from the implementation point of view) have a mechanism to generate the noise samples. Guided by this principle, we illustrate the overall robust training framework in Fig. 4, which consists of the following four stages. Roughly speaking, Stage 1 and Stage 2 are devoted to simulating the predictable noise, providing a differentiable network for modeling the distribution $P(\boldsymbol{\tau})$. Stage 3 deals with the conditional distribution $P(\boldsymbol{\xi}|\boldsymbol{\tau})$ through mimicking the unseen noise with an adversarial noise generation strategy. Eventually, Stage 4 handles the actual robust training of the image forgery detector $f_{\boldsymbol{\theta}}$ by using (8).

A. Modeling the Distribution $P(\boldsymbol{\tau})$

We now model the distribution $P(\boldsymbol{\tau})$, where the degradation is caused by the lossy operations of OSN platforms. From Section II-B, we know that the dominating degradation source of $\boldsymbol{\tau}$ is the applied JPEG compression, and the post-processing (e.g., enhancement filtering) and/or the possible downsampling also partially contribute to $\boldsymbol{\tau}$. For an image $\mathbf{x}_i$ and a *fixed* OSN platform, the incurred noise can be easily calculated by

$$\boldsymbol{\tau}_i = \mathrm{OSN}(\mathbf{x}_i) - \mathbf{x}_i, \tag{9}$$

where the function OSN$(\cdot)$ reflects all the operations conducted by the given OSN platform. Note that $\boldsymbol{\tau}_i$ depends on $\mathbf{x}_i$, namely, the noise is signal dependent. Seemingly, in this way, we can generate a lot of noise samples, which can be used to model the distribution of $P(\boldsymbol{\tau})$. However, in practice, such a naive modeling scheme is quite problematic. The processed image OSN$(\mathbf{x}_i)$ has to be obtained by uploading $\mathbf{x}_i$ to the specific OSN platform, and then downloading. Such procedure, on one hand, is time-consuming; on the other hand, many OSNs do not allow too many times of uploading/downloading operations. Some OSN platforms such as Weibo even ban the account if too many uploading/downloading operations are observed in a short period of time. This seriously limits the number of obtained noise samples, making such a naive scheme highly ineffective in practice.

To resolve this challenge, we resort to another strategy of modeling $P(\boldsymbol{\tau})$ in an inexplicit manner. We propose to use a substitute deep network for mimicking the OSN operations, so as to conveniently produce a large number of noise samples $\boldsymbol{\tau}_i$. Specifically, to be consistent with the image processing pipeline in the OSN platform, we train a DNN model, which explicitly embeds a differentiable layer to describe the JPEG compression. For an input image $\mathbf{x}_i$, we aim to learn a mapping $g_{\boldsymbol{\phi}} : \mathbb{R}^d \rightarrow \mathbb{R}^d$, where $g_{\boldsymbol{\phi}}$ is a network with trainable

parameters $\boldsymbol{\phi}$, predicting the OSN output. We employ a U-Net architecture for $g_{\boldsymbol{\phi}}$, as it is essentially an image-to-image mapping. The training procedure is illustrated in Stage 1 of Fig. 4, and then the well-trained $g_{\boldsymbol{\phi}^*}$ is employed in Stage 2 for modeling $P(\boldsymbol{\tau})$. At the training stage, we collect pairs of input image $\mathbf{x}_i \in \mathbb{R}^d$ and the OSN transmitted version $\text{OSN}(\mathbf{x}_i) \in \mathbb{R}^d$ in an offline manner. The objective function for training model $g_{\boldsymbol{\phi}}$ can be formulated as

$$\min_{\boldsymbol{\phi}} \Big\{ \mathcal{L}_r(g_{\boldsymbol{\phi}}(\mathbf{x}_i), \text{OSN}(\mathbf{x}_i)) \Big\}, \tag{10}$$

where $\mathcal{L}_r(\cdot,\cdot)$ measures the reconstruction loss defined by

$$\mathcal{L}_r(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2. \tag{11}$$

As we are more interested in learning the noise incurred by the OSN transmission rather than the processed image itself, we adopt a residual learning structure when designing $g_{\boldsymbol{\phi}}$. Bearing this in mind, we change the objective function into

$$\min_{\boldsymbol{\phi}} \Big\{ \mathcal{L}_r(\mathbf{x}_i + g_{\boldsymbol{\phi}}(\mathbf{x}_i), \text{OSN}(\mathbf{x}_i)) \Big\}. \tag{12}$$

As expected and will be verified experimentally, the residual learning is beneficial for the model optimization, significantly boosting the modeling performance.

In addition, we explicitly integrate a special layer into the model for better generating the structural, JPEG-like artifacts, which reflects the true situation in various OSN platforms. As a well-known fact, JPEG compression mainly consists of the following four steps: 1) color space transformation; 2) discrete cosine transform (DCT); 3) quantization; and 4) entropy coding. To enable the end-to-end optimization of the objective function in (12), we need to ensure that every step remains differentiable. Among the four steps, the quantization is the only non-differentiable one, mainly because the employed rounding function $\lfloor \cdot \rceil$ has 0 derivative everywhere. To have a differentiable quantization step, we approximate the rounding function with a differentiable version [29]:

$$\lfloor x \rceil_a = \lfloor x \rceil + (x - \lfloor x \rceil)^3, \tag{13}$$

where the maximum discrepancy 0.125 occurs at rounding 0.5. Upon having a differentiable JPEG layer, the objective function for training $g_{\boldsymbol{\phi}}$ becomes

$$\min_{\boldsymbol{\phi}} \mathcal{L}_r(\mathcal{J}_q(\mathbf{x}_i + g_{\boldsymbol{\phi}}(\mathbf{x}_i)), \text{OSN}(\mathbf{x}_i)), \tag{14}$$

where $\mathcal{J}_q$ represents the differentiable JPEG layer with a given QF q. In our training, q is uniformly sampled from the observed range in Fig. 2 (a). It is then straightforward to derive the noise $\boldsymbol{\tau}_i$ as

$$\boldsymbol{\tau}_i(q) = \mathcal{J}_q(\mathbf{x}_i + g_{\boldsymbol{\phi}^*}(\mathbf{x}_i)) - \mathbf{x}_i. \tag{15}$$

where $\boldsymbol{\phi}^*$ is obtained by solving the optimization problem (14) and q is the QF associated with the JPEG compression. For a given input $\mathbf{x}_i$, noticing the fact that q could change across OSN platforms, we can define the set of possible outcomes of $\boldsymbol{\tau}_i$ as

$$\Omega_{\boldsymbol{\tau}_i} = \{\boldsymbol{\tau}_i(q_1), \boldsymbol{\tau}_i(q_2), \cdots\}, \tag{16}$$

where $q_1, q_2, \cdots$ represent the underlying QF values adopted by OSNs. In our implementation, the QF values range from 71 to 95, as adopted by Facebook. Monte Carlo (MC) sampling scheme can then be easily implemented to generate a large number of noise samples for modeling the distribution $P(\boldsymbol{\tau})$.

Remark: In our noise modeling scheme, we dynamically sample the QF during the training, and hence simulate a universal network to mimic the general behavior of OSN platforms. Alternatively, we could train a specific network $g_{\boldsymbol{\phi}}$ for each individual q, so as to more accurately imitate the predictable noise. This idea is similar to that of some existing denoising networks, which train a network for each possible noise level [30]. However, we experimentally find that such alternative fails to bring noticeable improvements; meanwhile, it significantly increases the training cost, and makes it cumbersome to use in practice.

B. Modeling the Conditional Distribution of $\boldsymbol{\xi}$

In this subsection, we tackle the issue of modeling the conditional distribution $P(\boldsymbol{\xi}|\boldsymbol{\tau})$ so that we can solve the optimization problem in (8). The reason why we incorporate the noise term $\boldsymbol{\xi}$ is that the predictable noise $\boldsymbol{\tau}$ certainly cannot fully capture the noise behavior encountered in practice. For instance, different OSNs may adopt distinct processing procedures, e.g., adjusting the QF dynamically, performing resizing adaptively, or even introducing completely unseen/unknown operations.

A critical problem now is how to build a proper model for the unseen noise $\boldsymbol{\xi}$. Obviously, it is unrealistic to model the unseen noise $\boldsymbol{\xi}$ from the characteristic of the signal itself, as we do in Section IV-A. To resolve this challenge, we shift our position from the noise aspect to the detector $f_{\boldsymbol{\theta}}$, by studying the noise effect on the detection performance. Among the various underlying unseen noise $\boldsymbol{\xi}$, we actually only need to pay attention to the ones that degrade the detection performance, while neglecting those that have little effect on the detection. This motives us to employ a type of *adversarial noise* [20] when modeling $P(\boldsymbol{\xi}|\boldsymbol{\tau})$. Essentially, adversarial noises are generally imperceptible to the human senses while being able to cause severe model output errors. Meanwhile, the unseen noise $\boldsymbol{\xi}$ that we focus on is the one capable of fooling the detector and is also usually small (a highly distorted image would deviate from the purpose of making a forgery). Such similarity in terms of the effect to the detector $f_{\boldsymbol{\theta}}$ makes the adversarial noise a perfect candidate for modeling the noise $\boldsymbol{\xi}$.

From the adversarial point of view, there are various ways of defining the noise $\boldsymbol{\xi}$, as long as the adversarial example, created by adding the noise $\boldsymbol{\xi}$ to the original normal example, goes across the decision boundary. An illustrative example is given in Fig. 5, where the dotted lines indicate several possible directions for the adversarial noise. Noticing the fact that the noise $\boldsymbol{\xi}$ is typically of small amplitude, we propose to set the direction of $\boldsymbol{\xi}$ along the gradient of the cost function with respect to the input, so as to minimize the noise energy (see the red dotted line in Fig. 5). Therefore, for a given input $\mathbf{x}_i$, the predictable noise $\boldsymbol{\tau}_i$, and the target output $\mathbf{y}_i$, the unseen

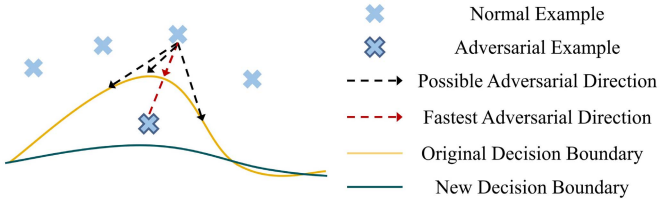

Fig. 5. An illustrative example of the learned decision boundary with or without the adversarial noise.

noise $\boldsymbol{\xi}_i$ is formulated as

$$\boldsymbol{\xi}_i = \mathcal{S}(\nabla_{\mathbf{x}_i}\mathcal{L}_b(f_{\boldsymbol{\theta}}(\mathbf{x}_i + \boldsymbol{\tau}_i), \mathbf{y}_i)), \tag{17}$$

where

$$\nabla_{\mathbf{x}_i}\mathcal{L}_b(f_{\boldsymbol{\theta}}(\mathbf{x}_i + \boldsymbol{\tau}_i), \mathbf{y}_i) = \frac{\partial \mathcal{L}_b(f_{\boldsymbol{\theta}}(\mathbf{x}_i + \boldsymbol{\tau}_i), \mathbf{y}_i)}{\partial \mathbf{x}_i} \tag{18}$$

calculates the gradient of the cost function $\mathcal{L}_b$ with respect to the input $\mathbf{x}_i$, and $\mathcal{S}$ returns the sign of the gradient. By adding such adversarial noises during the training, it is expected to make the learned model robust against not only the specific adversarial noise but also more general unseen noise.

However, the noise calculated by (17) depends on the specific input $\mathbf{x}_i$, rather than a general one applicable to all the examples in the training set and unknown examples. For comprehensively enhancing the generalization ability of the detector, we propose to adjust the direction of the adversarial noise to a global gradient direction. In this case, another crucial problem arising is how to accurately calculate the global gradient in an efficient manner. To this end, we adopt a strategy similar to the Stochastic Gradient Descend (SGD) [31], by a stochastic approximation approach from randomly selected subsets of the training dataset. More specifically, for the $(t+1)$-th input $\mathbf{x}_{t+1}$, the $\boldsymbol{\xi}_{t+1}$ (conditioning on $\boldsymbol{\tau}$) could be set as the average gradient calculated from the first t inputs, namely,

$$\boldsymbol{\xi}_{t+1} = \frac{1}{t}\sum_{i=0}^{t}\mathcal{S}(\nabla_{\mathbf{x}_i}\mathcal{L}_b(f_{\boldsymbol{\theta}}(\mathbf{x}_i + \boldsymbol{\tau}_i + \boldsymbol{\xi}_i), \mathbf{y}_i)), \tag{19}$$

where $\boldsymbol{\xi}_0$ is initialized as $\mathbf{0}$. Although (19) can be used to estimate the average gradients, it only reflects the gradients of specific known data (the training data), thus losing the generality. To alleviate the aforementioned problem and further improve the robustness, we propose to perturb the $\boldsymbol{\xi}_t$ in a small range. Here, it would be more ideal to use a parametric model to characterize the average gradients. To find an appropriate model for the average gradient, we first take a data-driven approach, analyzing the statistics of 1000 samples of $\boldsymbol{\xi}$ that are randomly selected from the training process. In Fig. 6, we visualize these 1000 random samples in a 2D space by using the t-SNE [32]. It can be seen that the sample points are concentrated around a certain center, and gradually vanish when they move away from the center. This phenomenon suggests us to use a Gaussian distribution for modeling the average gradient, i.e.,

$$\boldsymbol{\xi}_{t+1}|\boldsymbol{\tau} \sim \mathcal{N}(\boldsymbol{u}_{t+1}, \sigma^2\mathbf{I}), \tag{20}$$

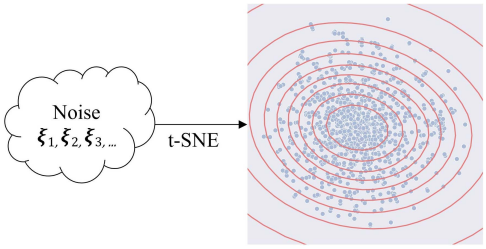

Fig. 6. Visualization of 1000 noise samples of $\boldsymbol{\xi}$ by using t-SNE [32].

where σ is an empirically set parameter for controlling the variance,

$$\boldsymbol{u}_{t+1} = \epsilon \cdot \frac{1}{t}\sum_{i=0}^{t}\mathcal{S}(\nabla_{\mathbf{x}_i}\mathcal{L}_b(f_{\boldsymbol{\theta}}(\mathbf{x}_i + \boldsymbol{\tau}_i + \boldsymbol{\xi}_i), \mathbf{y}_i)), \tag{21}$$

and ϵ is a parameter used for constraining the magnitude of the perturbations to avoid unnecessary model degradation. To be consistent with the real OSN data, we randomly sample ϵ from the observed distribution in Fig. 2 (b).

Upon having the parametric model of noise $\boldsymbol{\xi}_t|\boldsymbol{\tau}$, we can easily generate noise samples, and adopt an MC sampling method to conveniently calculate the average gradients in (19).

C. Details on the Robust Training

We now are ready to present the details on the optimization of the objective function given in (8). With the appropriate modeling techniques for the predictable noise $\boldsymbol{\tau}$ and the unseen noise $\boldsymbol{\xi}$, we can generate the noise samples, based on which we can approximate the objective function (8) using MC samples. More specifically, (8) can be expanded as

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{N}\sum_{j=1}^{m}\sum_{k=1}^{h}\mathcal{L}_b(f_{\boldsymbol{\theta}}(\mathbf{x}_i + \boldsymbol{\tau}_j + \boldsymbol{\xi}_k), \mathbf{y}_i). \tag{22}$$

where the expectations with respect to $\boldsymbol{\tau}$ and $\boldsymbol{\xi}$ are approximated with m and h MC samples, respectively. With this computable loss function, we are able to perform the robust training, as summarized in Algorithm 1. Let us briefly explain our algorithm for better understanding. In Algorithm 1, Lines 2∼7 are devoted to training a network $g_{\boldsymbol{\phi}^*}$ for estimating the predictable noise $\boldsymbol{\tau}$, which is used for Line 15. Line 16 utilizes the method proposed in Section IV-B to model the unseen noise $\boldsymbol{\xi}$ condition on $\boldsymbol{\tau}$. Then, in Line 18, the final objective function is computed and the parameters are updated in Lines 20∼22. Eventually, we produce the trained detector $f_{\boldsymbol{\theta}^*}$ in Line 26.

Remark: Our robust training scheme can also be regarded as a type of data augmentation technique, in which the injected noisy data are well designed according to the characteristics of OSN platforms and the detector itself. It is well-known that, in many cases, the improvements of robustness brought by the data augmentation often lead to the degraded performance of the *original* detector. It should be noted that such performance degradation occurs when the training and testing of the detector are based on the same data distribution. In reality, however, this is often not the case, as the testing data could be from vastly different distributions (the so-called data

Algorithm 1 The Training Algorithm

Input: Training data $\mathcal{D}_1$ and $\mathcal{D}_2$; training epochs N_1 and N_2; learning rates $l_{\boldsymbol{\phi}}$ and $l_{\boldsymbol{\theta}}$.
Output: Trained detector $f_{\boldsymbol{\theta}^*}$

Randomly initialize $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$
for *epoch = 1 to* N_1 **do**
  for *minibatch* $(\mathbf{x}_i, \mathbf{y}_i) \subset \mathcal{D}_2$ **do**
    $\mathbf{g}_{\boldsymbol{\phi}} = \nabla_{\boldsymbol{\phi}}[\mathcal{L}_r(\mathcal{J}_q(\mathbf{x}_i + g_{\boldsymbol{\phi}}(\mathbf{x}_i)), \mathbf{y}_i)]$ ▷ Eq. (14)
    $\boldsymbol{\phi} = \boldsymbol{\phi} - l_{\boldsymbol{\phi}} \cdot \mathbf{g}_{\boldsymbol{\phi}}$ ▷ Update $g_{\boldsymbol{\phi}}$
  end
end
Temporary output $g_{\boldsymbol{\phi}^*} = g_{\boldsymbol{\phi}}$
Initialize $\boldsymbol{u}_0 = \mathbf{0}$
for *epoch = 1 to* N_2 **do**
  for *minibatch* $(\mathbf{x}_i, \mathbf{y}_i) \subset \mathcal{D}_2$ **do**
    Initialize $\mathbf{L}_0 = \mathbf{0}$
    for *j = 1 to m* **do**
      $q_j \sim \text{Uniform}(71, 95)$ ▷ Sample QF
      $\boldsymbol{\tau}_j = \mathcal{J}_{q_j}(\mathbf{x}_i + g_{\boldsymbol{\phi}^*}(\mathbf{x}_i)) - \mathbf{x}_i$ ▷ Model $\boldsymbol{\tau}$
      $\{\boldsymbol{\xi}_1, \cdots, \boldsymbol{\xi}_h\} \sim \mathcal{N}(\boldsymbol{u}_{i-1}, \sigma^2\mathbf{I})$ ▷ Model $\boldsymbol{\xi}$
      $\boldsymbol{\xi}_k = \min(\max(\boldsymbol{\xi}_k, \epsilon), -\epsilon)$
      $\mathbf{L}_j = \mathbf{L}_{j-1} + \sum_{k=1}^{h} \mathcal{L}_b(f_{\boldsymbol{\theta}}(\mathbf{x}_i + \boldsymbol{\tau}_j + \boldsymbol{\xi}_k), \mathbf{y}_i)$ ▷ Eq. (22)
    end
    $\mathbf{g}_{\boldsymbol{\theta}} = \nabla_{\boldsymbol{\theta}}\mathbf{L}_m$
    $\mathbf{g}_{\mathbf{x}_i} = \nabla_{\mathbf{x}_i}\mathbf{L}_m$
    $\boldsymbol{\theta} = \boldsymbol{\theta} - l_{\boldsymbol{\theta}} \cdot \mathbf{g}_{\boldsymbol{\theta}}$ ▷ Update $f_{\boldsymbol{\theta}}$
    $\boldsymbol{u}_i = \boldsymbol{u}_{i-1} + \epsilon \cdot \mathcal{S}(\mathbf{g}_{\mathbf{x}_i})$ ▷ Eq. (21)
  end
end
Final output $f_{\boldsymbol{\theta}^*} = f_{\boldsymbol{\theta}}$

bias [33]). From this perspective, it is very crucial to consider the generalization of the detector, in which case, the data augmentation usually has a positive effect [34]. As expected and will be verified experimentally, our proposed scheme can effectively boost the robustness against the transmission over various OSN platforms. Meanwhile, compared with the baseline method without OSN transmissions, the proposed robust version also outperforms it by a big margin.

V. Experimental Results

In this section, we present the experimental results to show the superior performance of our proposed method. For better presentation, we first explain the detailed experimental settings. The detection results over four publicly available datasets against four popular OSNs are then provided. For comparison purposes, we also provide the results of four state-of-the-art methods. Finally, extensive ablation experiments are conducted and further discussions on more challenging real-world cases are given.

A. Experimental Setup

1) Training/Validation Datasets: For the training of the OSN network $g_{\boldsymbol{\phi}}$, a recently released dataset `WEI` (denoted as $\mathcal{D}_1$) [18] is adopted. The `WEI` dataset contains over 1300 original images and their processed versions upon the transmission over Facebook. It should be noted that we only use the data from Facebook for training the network $g_{\boldsymbol{\phi}}$. On one hand, it was mentioned in [18] that the operations in Facebook are very diverse, and may be treated as a superset of those operations conducted in other OSN platforms. This implies that the network trained with Facebook data could have desirable generalizability to other OSN platforms or even to some unknown (new) operations. On the other hand, training with only one particular OSN data could significantly simplify the task of collecting the training data, and hence, provide convenience from the implementation perspective.

For the training of the baseline detector $f_{\boldsymbol{\theta}}$, similar to [1], [35], we use the `Dresden` [36] dataset as the source of pristine images. We then generate the forged images by splicing the pristine images with the objects from the `MS-COCO` [37] dataset. The dataset of these forged images is denoted as $\mathcal{D}_2$.

The above datasets $\mathcal{D}_1$ and $\mathcal{D}_2$ are further randomly divided into training and validation sets with the ratio of 9 : 1.

2) Testing Datasets: The following four widely-used datasets are adopted for the performance evaluations.

- `DSO` [23]: This dataset is formed by 100 skillfully-forged images, with the resolution of 2048 × 1536. To improve photorealism, these forged images undergo a series of post-processing, e.g., adjustments of color and illumination.
- `Columbia` [21]: This dataset provides 160 splicing forgeries, where the source images are captured by four different cameras. These forged images are uncompressed and of high quality, with resolutions ranging from 757 × 568 to 1152 × 768.
- `NIST` [24]: This dataset contains 564 high-resolution forgeries that are manipulated by commonly used tampering operations, e.g., splicing, removal and copy-move, and post-processing with unknown editing software. The resolutions of the forgeries range from 500 × 500 to 5616 × 3744.
- `CASIA` [22]: This dataset has 920 forgeries created by splicing with Adobe Photoshop CS3 version 10.0.1 on Windows XP. All images are resized to 384 × 256 and are in JPEG format.

In addition, to evaluate the robustness of image forgery detection methods against the OSN transmission, we create an OSN-transmitted dataset by including the OSN-transmitted versions of the above four datasets. More specifically, we transmit all these forgeries through the four most popular OSNs (Facebook, Whatsapp, Weibo, and Wechat), and eventually obtain our OSN-transmitted dataset with 6976 forgeries in total. When producing this OSN-transmitted dataset, we use a ThinkPad X1 Carbon Gen 8 laptop of Windows 10 21H1 operating system to perform the image uploading and downloading via the Facebook, Weibo, and Whatsapp platforms, and an iPhone 12 of iOS 14.1 for the Wechat platform. To better mimic the sharing process of ordinary users in practice, we adopt the default settings for all the platforms. For instance, the Weibo transmitted images contain automatically generated watermarks.

TABLE I

QUANTITATIVE COMPARISONS BY USING AUC, F1, AND IOU AS CRITERIA. FOR EACH COLUMN WITHIN THE SAME OSN TRANSMISSION, THE HIGHEST VALUE IS **BOLD**, AND "-" INDICATES NOT APPLICABLE

Models	OSNs	Test Datasets														
		DSO [23]			Columbia [21]			NIST [24]			CASIA [22]			Average		
		AUC	F1	IoU	AUC	F1	IoU	AUC	F1	IoU	AUC	F1	IoU	AUC	F1	IoU
MT-Net [14]	-	.795	.344	.253	.747	.357	.258	.634	.088	.054	.776	.130	.086	.738	.230	.163
NoiPri [16]	-	**.902**	.339	.253	.840	.362	.260	.672	.119	.078	-	-	-	.804	.273	.197
ForSim [15]	-	.796	**.487**	**.371**	.731	.604	.474	.642	.188	.123	.554	.169	.102	.681	.362	.268
DFCN [1]	-	.724	.303	.227	.789	.541	.395	.778	.250	.204	.654	.192	.119	.736	.322	.236
Baseline	-	.761	.312	.194	.763	.616	.501	.682	.221	.139	.774	.402	.342	.745	.388	.294
Ours	-	.854	.436	.308	**.862**	**.707**	**.608**	**.783**	**.332**	**.255**	**.873**	**.509**	**.465**	**.843**	**.496**	**.409**
MT-Net [14]	Facebook	.638	.109	.071	.626	.103	.056	.652	.095	.057	.763	.102	.065	.670	.102	.062
NoiPri [16]	Facebook	.777	.150	.097	.722	.223	.143	.583	.057	.034	-	-	-	.694	.143	.091
ForSim [15]	Facebook	.689	.356	.238	.607	.450	.304	.580	.140	.085	.537	.157	.094	.603	.276	.180
DFCN [1]	Facebook	.673	.238	.184	.687	.479	.338	.705	.207	.138	.654	.190	.116	.680	.278	.194
Baseline	Facebook	.714	.180	.105	.689	.594	.497	.646	.200	.136	.728	.350	.298	.694	.331	.259
Ours	Facebook	**.859**	**.447**	**.320**	**.883**	**.714**	**.611**	**.783**	**.329**	**.253**	**.862**	**.462**	**.417**	**.847**	**.488**	**.400**
MT-Net [14]	Whatsapp	.616	.081	.052	.630	.098	.052	.702	.101	.062	.763	.099	.063	.678	.095	.057
NoiPri [16]	Whatsapp	.606	.081	.057	.705	.230	.148	.579	.073	.045	-	-	-	.630	.131	.083
ForSim [15]	Whatsapp	.542	.233	.139	.595	.436	.294	.586	.137	.082	.525	.151	.091	.562	.239	.152
DFCN [1]	Whatsapp	.645	.264	.162	.692	.471	.331	.689	.187	.125	.655	.191	.117	.670	.278	.184
Baseline	Whatsapp	.754	.097	.069	.692	.617	.526	.659	.207	.159	.751	.372	.321	.714	.323	.269
Ours	Whatsapp	**.839**	**.341**	**.233**	**.889**	**.727**	**.628**	**.785**	**.313**	**.239**	**.866**	**.478**	**.431**	**.845**	**.465**	**.383**
MT-Net [14]	Weibo	.606	.057	.036	.620	.103	.056	.671	.088	.053	.754	.099	.063	.663	.087	.052
NoiPri [16]	Weibo	.606	.093	.061	.664	.175	.108	.580	.054	.030	-	-	-	.616	.107	.066
ForSim [15]	Weibo	.568	.260	.165	.610	.453	.312	.581	.150	.094	.542	.165	.100	.575	.257	.168
DFCN [1]	Weibo	.639	.227	.140	.676	.458	.319	.706	.192	.125	.653	.191	.117	.668	.267	.175
Baseline	Weibo	.703	.120	.073	.681	.558	.477	.683	.163	.116	.762	.338	.310	.707	.294	.244
Ours	Weibo	**.808**	**.370**	**.253**	**.883**	**.724**	**.626**	**.780**	**.294**	**.219**	**.858**	**.466**	**.421**	**.832**	**.463**	**.380**
MT-Net [14]	Wechat	.582	.076	.045	.613	.199	.125	.654	.095	.057	.724	.080	.048	.643	.113	.069
NoiPri [16]	Wechat	.618	.098	.062	.639	.202	.124	.575	.041	.026	-	-	-	.610	.114	.070
ForSim [15]	Wechat	.564	.247	.147	.650	.496	.354	.581	.136	.082	.532	.153	.091	.582	.258	.168
DFCN [1]	Wechat	.653	.221	.137	.676	.487	.344	.701	.176	.114	.651	.193	.119	.670	.269	.179
Baseline	Wechat	.668	.076	.051	.655	.535	.431	.626	.170	.128	.670	.182	.152	.655	.241	.191
Ours	Wechat	**.823**	**.366**	**.252**	**.883**	**.727**	**.631**	**.764**	**.286**	**.214**	**.833**	**.405**	**.358**	**.826**	**.446**	**.364**

It needs to be emphasized that there is **NO** overlap between the training and testing datasets, better simulating the real situation and evaluating the generalization of the forgery detection algorithms.

3) Comparative Methods: We compare our proposed scheme with the following four state-of-the-art forensic methods: MT-Net [14], NoiPri [16], ForSim [15], and DFCN [1]. For fairness, we compare our scheme with the DFCN *retrained* on our training dataset $\mathcal{D}_2$. For the other three competitors, since their detection mechanism relies on extracting abnormal regions or features in the image, i.e., no specific training samples of a forensic trace are required, we adopt their officially released models.

4) Evaluation Criteria: We adopt the following commonly used pixel-level metrics: the Area Under the receiver operating characteristic Curve (AUC), the F1-score (F1), and the Intersection over Union (IoU). For calculating the F1 and IoU scores, thresholding is necessary as the direct outputs of the network are probability values. Similar to [1], we set the threshold as 0.5.

5) Implementation Details: The proposed method is implemented using the PyTorch deep learning framework and adopting the Adam [38] with default parameters as the optimizer. The batch size is set to 32 and every epoch contains 312 batches. We train the network with an initial learning rate 1e-4, and halve it if the evaluation criteria fail to increase for 5 epochs until the convergence. All the images used in the training phase are cropped to 256×256, while there is no size limit for the testing phase. To embrace the concept of reproducible research, the code of our paper and the collected datasets are made available at **https://github.com/HighwayWu/ImageForensicsOSN**, serving as a useful resource to our research community for fighting against the OSN-transmitted forgeries.

B. Quantitative Comparisons

The quantitative comparisons in terms of the AUC, F1 and IoU (higher are better) in the pixel domain are presented in Table I. Here we also report the results of our baseline detector in Section III for demonstrating the improvement of our robust training scheme in a comparative way. As can be observed, when the forgeries are not transmitted through an OSN, the detection methods ForSim [15], DFCN [1] and ours achieve comparable results, while MT-Net [14] and NoiPri [16] perform slightly worse. It should be noted that, NoiPri cannot be applied to detect the forgeries in CASIA due to their small resolutions, while our method has no such limitation and perform even better than the other competitors on CASIA.

In the scenario that the forgeries are passed through OSNs (Facebook, Whatsapp, Weibo and Wechat), the detection performance of all existing methods has deteriorated significantly. For instance, after the transmission over Facebook, Whatsapp, Weibo and Wechat, the IoU scores associated with MT-Net drop by 10.1%, 10.6%, 11.1%, and 9.4%, respectively, compared to the scenario without OSN transmission. Such severe degradation is probably due to the fact that the lossy operations

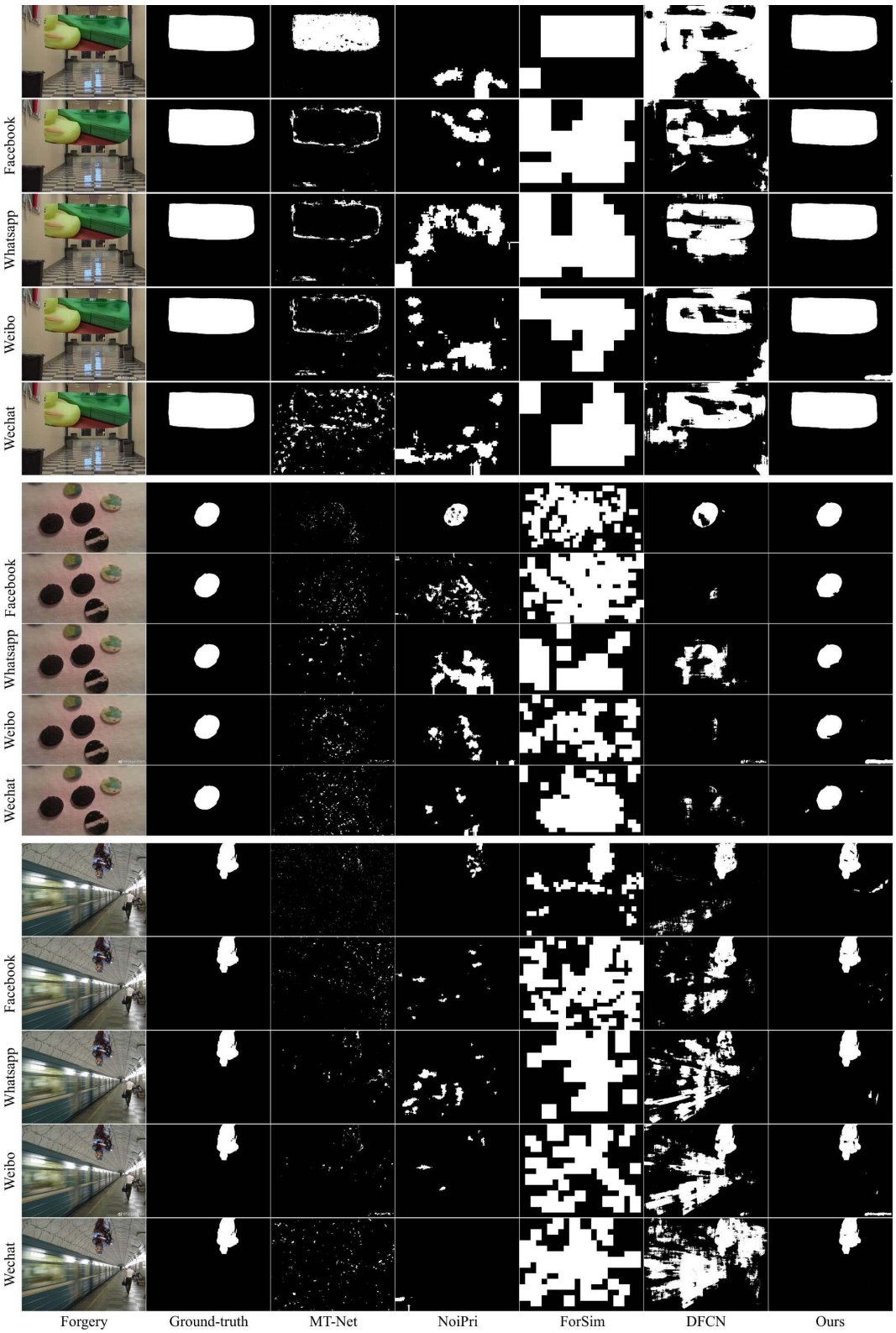

Fig. 7. Qualitative comparisons for detecting the OSN-transmitted forgeries. For each row, the images from left to right are forgery (input), ground-truth, detection result (output) generated by `MT-Net` [14], `NoiPri` [16], `ForSim` [15], `DFCN` [1] and ours, respectively. The forgeries in each group from top to bottom are the cases without OSN transmission, and with Facebook, Whatsapp, Weibo, and Wechat transmissions, respectively.

conducted by OSNs destroy a large portion of the forgery artifacts. In contrast, thanks to the appropriate noise modeling of $\boldsymbol{\tau}$ and $\boldsymbol{\xi}$, our proposed method exhibits rather desirable robustness against the OSN transmissions and still leads to accurate forgery detections. Taking Facebook for example, the IoU reduction is only 0.9%. It can also be noticed that the degradations of the forgery detection performance are slightly larger for Whatsapp, Weibo and Wechat, with IoU

TABLE II

FALSE POSITIVE RATE ANALYSIS WITH THE THRESHOLD *TH*

TH	OSNs	MT-Net	NoiPri	ForSim	DFCN	Ours
0.5	-	**.019**	.060	.051	.022	.024
	Facebook	**.024**	.105	.052	.045	.027
0.6	-	**.015**	.027	.051	.018	.019
	Facebook	**.017**	.045	.052	.035	.019
0.7	-	.013	**.012**	.051	.014	.013
	Facebook	**.011**	.018	.052	.027	**.011**
0.8	-	.011	**.005**	.051	.011	.008
	Facebook	.007	.006	.052	.019	**.005**
0.9	-	.004	**.001**	.051	.006	.003
	Facebook	.003	.002	.052	.010	**.001**
0.95	-	**.000**	**.000**	.051	.001	**.000**
	Facebook	**.000**	**.000**	.052	**.000**	**.000**

reductions being 2.6%, 2.9% and 4.5%, respectively. This is mainly because, compared with Facebook, Whatsapp, Weibo and Wechat adopt more stringent compressions for uploaded images, causing more evidence loss. In addition, for training our method, we only use the Facebook data, without any Whatsapp, Weibo or Wechat data at all. From Table I, we can see the scheme trained by using Facebook data can generalize well to Whatsapp, Weibo and Wechat transmitted images.

C. Qualitative Comparisons

In addition to the quantitative comparisons, we also compare different methods qualitatively, as shown in Fig. 7. More specifically, Fig. 7 gives several representative examples from the testing datasets (Columbia [21] and NIST [24]). It can be seen that in the normal case (no OSN transmission), the existing detection methods perform relatively well, e.g., the MT-Net and ForSim in the first case, and the NoiPri and DFCN in the second case. However, these methods cannot achieve satisfactory detection performance in the cases of OSN transmitted versions. Take NoiPri in the second case for example. For Facebook, Whatsapp, Weibo and Wechat transmitted images, the identified forged regions also spread over several objects, making the forgery detection results much less useful. In contrast, our proposed method can learn more robust forgery features, and thereby generate more precise detection results over these challenging cases, primarily thanks to the robust training scheme with the compound noise modeling. Note that Weibo will automatically generate a visible watermark in the lower right corner of the uploaded image. Although the watermark is essentially a kind of forgery (and our method can correctly detect it), we still use the original ground-truth masks for evaluating the detection performance of all the methods.

D. False Positive Rate Analysis

In practical scenarios, it is also important to evaluate image forgery detection methods in terms of false positive rate (FPR). This is because forged images only occupy a very small portion in the majority of applications, and hence, good forgery detection methods are expected to have low FPR [4]. To this end, we additionally measure the pixel-level FPR of our scheme and the competing schemes on the VISION dataset [39], which contains the authentic images (and their Facebook-transmitted versions) captured using 35 different portable devices. The results are compiled in Table II, where the parameter TH is used to threshold the probability output. Note that ForSim does not need to have the thresholding process. As can be observed, our method leads to small FPR (less than 3%), and the FPR performance can be improved with the increasing TH. In fact, except ForSim, the remaining four methods result in relatively comparable FPR performance. Also, Facebook transmission just slightly affects the FPR performance of all methods. It should also be emphasized that the parameter TH trades off the FPR and the false negative rate (FNR); namely, reducing FPR by increasing TH produces higher FNR.

TABLE III

ABLATION STUDIES REGARDING THE TRAINING OF OSN NETWORK. $^-$LOWER IS BETTER. $^+$HIGHER IS BETTER

Network g_{ϕ}	PSNR$^+$	SSIM$^+$	MSE$^-$
U-Net	32.22	0.3968	48.60
+ Res.	43.39	0.8699	3.31
+ JPEG	**44.74**	**0.9270**	**2.77**

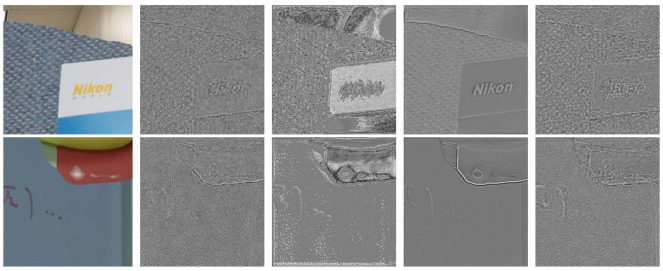

Fig. 8. Visualization of the predictable noise $\boldsymbol{\tau}$ regarding different network architectures.

E. Ablation Studies

1) The Training of OSN Network: Before starting the ablation studies of our modelled noise $\boldsymbol{\tau}$ and $\boldsymbol{\xi}$, we first present how different architectures of the OSN network $g_{\boldsymbol{\phi}}$ affect the simulation of the predictable noise $\boldsymbol{\tau}$. For simplicity, we denote the vanilla network $g_{\boldsymbol{\phi}}$ as "U-Net", the network with residual learning as "+ Res.", and the network with both residual learning and differentiable JPEG layer as "+ JPEG". The objective functions corresponding to these three alternatives have been given in (10), (12), and (14), respectively. The quantitative comparisons in terms of the PSNR, SSIM (higher are better) and MSE (lower is better) in the residual domain are reported in Table III, where some predictable noises $\boldsymbol{\tau}$ are visualized in Fig. 8 for the comparison. As can be observed, the residual learning is very effective in making the network focus on predicting the OSN noise, rather than the image content, and hence achieves a large PSNR gain of 11.17 dB. However, the predictable $\boldsymbol{\tau}$ in this case is not visually similar to the real OSN noise. This may be because it is challenging for a standard convolutional neural network to generate the unique JPEG-like artifacts. Upon the implicit integration of the differentiable JPEG layer into the network $g_{\boldsymbol{\phi}}$, the predictable noise not only can be further improved (PSNR gains 1.35 dB),

TABLE IV

ABLATION STUDIES REGARDING THE MODELING OF THE PREDICTABLE NOISE τ AND UNSEEN NOISE ξ. VALUES IN BRACKETS REPRESENT THE DIFFERENCE WITH THE CORRESPONDING BASELINE DETECTOR

Detector f_θ	Test Datasets					
	Trans. w/o OSN			Trans. w/ Facebook		
	AUC	F1	IOU	AUC	F1	IOU
#1 SE-U-Net (Baseline)	.745	.388	.294	.694	.331	.259
#2 SE-U-Net + τ	.755 (+.010)	.400 (+.012)	.325 (+.031)	.733 (+.039)	.377 (+.046)	.311 (+.052)
#3 SE-U-Net + ξ by FGSM [40]	.773 (+.028)	.402 (+.014)	.354 (+.060)	.729 (+.035)	.350 (+.031)	.319 (+.060)
#4 SE-U-Net + ξ in (20)	.794 (+.049)	.471 (+.083)	.383 (+.089)	.753 (+.059)	.417 (+.086)	.340 (+.081)
#5 SE-U-Net + τ + ξ in (20)	.843 (+.098)	.496 (+.108)	.409 (+.115)	.847 (+.153)	.488 (+.157)	.400 (+.141)
#6 DPN	.719	.319	.224	.651	.208	.135
#7 DPN + τ + ξ in (20)	.778 (+.059)	.421 (+.102)	.350 (+.126)	.776 (+.125)	.449 (+.241)	.385 (+.250)

but also more resembles the ground-truth visually. Also, the interpretability of the whole OSN network g_ϕ now becomes clear.

2) The Adoption of Modeling τ and ξ: We now conduct the ablation studies of our proposed training scheme by analyzing how each modelled noise (i.e., the predictable noise τ and the unseen noise ξ) contributes to the final detection performance. To this end, we first prohibit the use of each noise in the scheme, and then evaluate the performance of different retrained detectors with appropriate settings. The obtained results are shown in Table IV.

As can be seen, introducing the predictable noise τ in the training of detector (#2 row) can slightly improve the detection performance (e.g., 1.2% gains in F1), which is more obvious in the case of Facebook transmission (e.g., 4.6% gains in F1). However, since it is incomplete to only adopt the predictable noise τ, as mentioned in Section IV-B, we further involve the designed unseen noise ξ. The results in #4 row imply that ξ can effectively enhance the robustness of the detector, bringing a more significant improvement (e.g., 8.6% gains in F1). Finally, #5 row demonstrates that when the compound noise τ and ξ are applied simultaneously, the detector can be much more robust to the target environment, which is crucial for the forgery detection task over OSN transmission (e.g. 15.7% gains in F1).

In addition to the aforementioned way of defining the unseen noise ξ, we illustrate another design methodology, i.e., using the FGSM [40]. The comparison (#3 row) shows that the noise defined by FGSM cannot effectively improve the overall robustness of the detector, indicating that our proposed noise modeling scheme for the unseen noise ξ is highly non-trivial.

Finally, instead of only using the SE-U-Net as the detector, we adopt another well-known architecture, DPN [41], to show the versatility of our proposed training scheme. As shown in rows #6 and #7, the robustness of the DPN can also be well strengthened by our robust training method.

F. Some Further Robustness Evaluations

Although the proposed model is mainly designed to counter the lossy operations conducted by OSNs, we would also like to evaluate its robustness under some more commonly used degradation scenarios, such as noise addition, cropping, resizing, blurring, and standalone JPEG compression. Such

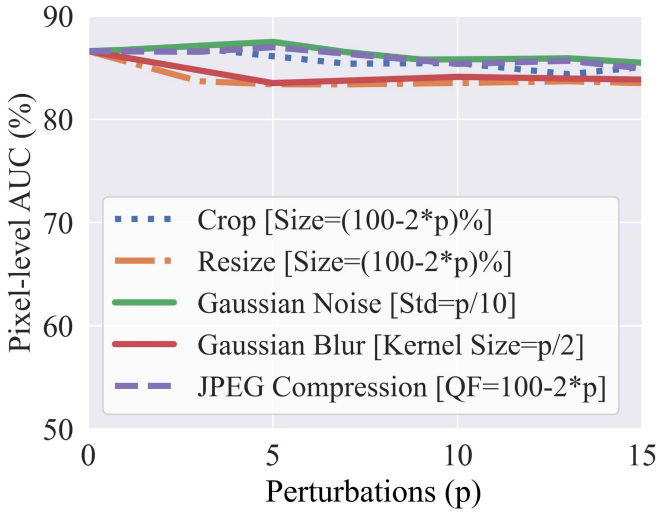

Fig. 9. Robustness evaluations of our proposed model against cropping, resizing, blurring, noising and JPEG compression.

evaluation is very critical in real-world cases because these types of post-processing operations are often adopted to erase or conceal the forged artifacts. To this end, we apply these post-processing operations to the original test set `Columbia` and report the quantitative results in Fig. 9. For the convenience of demonstration, we utilize a unified parameter p for controlling the magnitudes of different operations, e.g., for the Gaussian noise, $p/10$ stands for the standard deviation, while $100 - 2 * p$ represents the QF employed in the JPEG compression. The origin of the horizontal axis ($p = 0$) corresponds to the case without any post-processing. As can be observed, the overall performance is rather consistent with the increase of the perturbation intensity. More specifically, the detection performance remains almost unchanged for the cases of center cropping, adding Gaussian noise or performing JPEG compression within the given QF range. For the resizing and Gaussian blurring, the performance drops are slightly increased to approximately 3%. The above evaluation results indicate that our proposed model exhibits desirable robustness against these commonly used post-processing operations as well.

Furthermore, we measure the robustness of our proposed model under more challenging scenarios, namely, re-transmission or cross-transmission over OSNs (e.g., images

TABLE V
DETECTION AGAINST CROSS TRANSMISSION ON DATASET Columbia

OSNs 1st	OSNs 2nd	AUC	F1	IOU	Avg. File Size
-	-	.862	.707	.608	1713 KB
Facebook	-	.883	.714	.611	87 KB
Facebook	Facebook	.883	.714	.611	87 KB
Facebook	Weibo	.849	.674	.552	77 KB
Weibo	-	.883	.724	.626	83 KB
Weibo	Weibo	.882	.721	.621	83 KB
Weibo	Facebook	.883	.724	.626	83 KB

are downloaded and re-uploaded to the same or different OSNs), which may happen quite often in reality. Specifically, we consider the transmission via Facebook followed by Facebook/Weibo, and Weibo followed by Weibo/Facebook. The detection results in Table V demonstrate rather desirable robustness against the second round of OSN transmissions. An interesting phenomenon that should be pointed out is: after being initially processed by Facebook or Weibo, the forgeries will not be further compressed by Facebook (see the last column). The reason may be that Facebook will not take additional actions on the images that have already met the size condition or quality constraints, which brings convenience to the forensic tasks. However, cross-transmission over different OSNs does aggravate the quality loss of the images (e.g., transmission to Facebook followed by Weibo), thereby slightly degrading the eventual detection performance.

G. Discussions

Before ending this section, we make a supplementary explanation to the training of the OSN network $g_{\boldsymbol{\phi}}$. The OSN used to make the training set $\mathcal{D}_1$ is not necessarily Facebook, and other platforms such as Weibo and/or Wechat can be adopted as well. In fact, we have also tried to use the data transmitted over Weibo/Wechat for the training of $g_{\boldsymbol{\phi}}$, and observed similar detection performance as given in Table I. The reason behind this may be two-fold: 1) different OSNs adopt some common operations, e.g., the JPEG compression, which leads to similar predictable noise $\boldsymbol{\tau}$ learned by the network $g_{\boldsymbol{\phi}}$; and 2) the unseen noise $\boldsymbol{\xi}$ can dynamically fine-tune the predictable $\boldsymbol{\tau}$, enabling the detector $f_{\boldsymbol{\theta}}$ to learn how to improve its robustness.

VI. CONCLUSION

In this paper, we propose a novel training scheme for improving the robustness of the image forgery detection against various OSN-based transmissions. The proposed scheme is designed with the assistance of the modeling of a predictable noise $\boldsymbol{\tau}$ as well as an intentionally introduced unseen noise $\boldsymbol{\xi}$. Experimental results are provided to demonstrate the superiority of our scheme compared with several state-of-the-art methods. Further, we build an OSN-transmitted forgery dataset for future research of the forensic community.

As the future work, we would extend the proposed robust training scheme to deal with more complex degradation scenarios, such as screen capturing, printing and re-photographing, etc. Additionally, we will investigate whether an image restoration network can be used to assist the forgery detection in severely degraded scenarios.

ACKNOWLEDGMENT

This work was initially inspired by the Security AI Challenge: Forgery Detection on Certificate Image, Alibaba Security.

REFERENCES

[1] P. Zhuang, H. Li, S. Tan, B. Li, and J. Huang, "Image tampering localization using a dense fully convolutional network," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 2986–2999, 2021.
[2] S. Lyu, X. Pan, and X. Zhang, "Exposing region splicing forgeries with blind local noise estimation," *Int. J. Comput. Vis.*, vol. 110, no. 2, pp. 202–221, Nov. 2014.
[3] Y. Li and J. Zhou, "Fast and effective image copy-move forgery detection via hierarchical feature point matching," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 5, pp. 1307–1322, May 2019.
[4] X. Kang, M. C. Stamm, A. Peng, and K. J. R. Liu, "Robust median filtering forensics using an autoregressive model," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 9, pp. 1456–1468, Sep. 2013.
[5] H. Li, W. Luo, and J. Huang, "Localization of diffusion-based inpainting in digital images," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 12, pp. 3050–3064, Dec. 2017.
[6] M. Huh, A. Liu, A. Owens, and A. A. Efros, "Fighting fake news: Image splice detection via learned self-consistency," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 101–117.
[7] J.-L. Zhong and C.-M. Pun, "An end-to-end dense-InceptionNet for image copy-move forgery detection," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 2134–2146, 2020.
[8] J. Chen, X. Kang, Y. Liu, and Z. J. Wang, "Median filtering forensics based on convolutional neural networks," *IEEE Signal Process. Lett.*, vol. 22, no. 11, pp. 1849–1853, Nov. 2015.
[9] H. Wu and J. Zhou, "IID-Net: Image inpainting detection network via neural architecture search and attention," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Apr. 22, 2021, doi: 10.1109/TCSVT.2021.3075039.
[10] A. Li *et al.*, "Noise doesn't lie: Towards universal detection of deep inpainting," in *Proc. 13th Int. Joint Conf. Artif. Intell.*, Aug. 2021, pp. 1–7.
[11] D. Cozzolino, G. Poggi, and L. Verdoliva, "Splicebuster: A new blind image splicing detector," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Nov. 2015, pp. 1–6.
[12] L. Bondi, S. Lameri, D. Guera, P. Bestagini, E. J. Delp, and S. Tubaro, "Tampering detection and localization through clustering of camera-based CNN features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1855–1864.
[13] B. Bayar and M. C. Stamm, "Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 11, pp. 2691–2706, Nov. 2018.
[14] Y. Wu, W. Abdalmageed, and P. Natarajan, "ManTra-Net: Manipulation tracing network for detection and localization of image forgeries with anomalous features," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9543–9552.
[15] O. Mayer and M. C. Stamm, "Forensic similarity for digital images," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1331–1346, 2020.
[16] D. Cozzolino and L. Verdoliva, "Noiseprint: A CNN-based camera model fingerprint," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 114–159, 2020.
[17] W. Sun, J. Zhou, R. Lyu, and S. Zhu, "Processing-aware privacy-preserving photo sharing over online social networks," in *Proc. 24th ACM Int. Conf. Multimedia*, Oct. 2016, pp. 581–585.
[18] W. Sun, J. Zhou, Y. Li, M. Cheung, and J. She, "Robust high-capacity watermarking over online social network shared images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 3, pp. 1208–1221, Mar. 2021.
[19] *Security Ai Competition: Forgery Detection on Certificate Image*. Accessed: Jan. 23, 2022. [Online]. Available: https://tianchi.aliyun.com/competition/entrance/531812/information
[20] C. Szegedy *et al.*, "Intriguing properties of neural networks," in *Proc. Int. Conf. Learn. Representat.*, 2014, pp. 1–10.

[21] Y.-F. Hsu and S.-F. Chang, "Detecting image splicing using geometry invariants and camera characteristics consistency," in *Proc. IEEE Int. Conf. Multimedia Expo.*, Jul. 2006, pp. 549–552.
[22] J. Dong, W. Wang, and T. Tan, "CASIA image tampering detection evaluation database," in *Proc. IEEE China Summit Int. Conf. Signal Inf. Process.*, Jul. 2013, pp. 422–426.
[23] T. J. de Carvalho, C. Riess, E. Angelopoulou, H. Pedrini, and A. Rocha, "Exposing digital image forgeries by illumination color classification," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 7, pp. 1182–1194, Jul. 2013.
[24] *Nist Nimble 2016 Datasets*. Accessed: Jan. 23, 2022. [Online]. Available: https://www.nist.gov/itl/iad/mig/nimble-challenge-2017-evaluation/
[25] W. Sun, J. Zhou, L. Dong, J. Tian, and J. Liu, "Optimal pre-filtering for improving Facebook shared images," *IEEE Trans. Image Process.*, vol. 30, pp. 6292–6306, 2021.
[26] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Int.* Switzerland: Springer, 2015, pp. 234–241.
[27] V. Nair and G. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 807–814.
[28] A. G. Roy, N. Navab, and C. Wachinger, "Recalibrating fully convolutional networks with spatial and channel 'squeeze and excitation' blocks," *IEEE Trans. Med. Imag.*, vol. 38, no. 2, pp. 540–549, 2018.
[29] R. Shin and D. Song, "Jpeg-resistant adversarial images," in *Proc. Neural Inf. Process. Syst. Workshop*, 2017, pp. 1–6.
[30] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian Denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.
[31] S. Ruder, "An overview of gradient descent optimization algorithms," 2016, *arXiv:1609.04747*.
[32] L. V. der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 1–27, 2008.
[33] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *Proc. CVPR*, Jun. 2011, pp. 1521–1528.
[34] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros, "CNN-generated images are surprisingly easy to Spot…for now," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8695–8704.
[35] J. H. Bappy, C. Simons, L. Nataraj, B. S. Manjunath, and A. K. Roy-Chowdhury, "Hybrid LSTM and encoder-decoder architecture for detection of image forgeries," *IEEE Trans. Image Process.*, vol. 28, no. 7, pp. 3286–3300, Jul. 2019.
[36] T. Gloe and R. Böhme, "The dresden image database for benchmarking digital image forensics," *J. Digit. Forensic Pract.*, vol. 3, nos. 2–4, pp. 150–159, 2010.
[37] T.-Y. Lin *et al.*, "Microsoft coco: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.
[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[39] D. Shullani, M. Fontani, M. Iuliani, O. A. Shaya, and A. Piva, "VISION: A video and image dataset for source identification," *EURASIP J. Inf. Secur.*, vol. 2017, no. 1, pp. 1–16, Dec. 2017.
[40] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. Int. Conf. Learn. Representat.*, 2015, pp. 1–11.
[41] Y. Chen, J. Li, H. Xiao, X. Jin, S. Yan, and J. Feng, "Dual path networks," in *Proc. Neural Info. Process. Syst.*, 2017, pp. 4470–4478.

Haiwei Wu (Student Member, IEEE) received the B.S. and M.S. degrees in computer science from the University of Macau, Macau, China, in 2018 and 2020, respectively, where he is currently pursuing the Ph.D. degree with the Department of Computer and Information Science, Faculty of Science and Technology. His research interests include multimedia security, image processing, and machine learning.

Jiantao Zhou (Senior Member, IEEE) received the B.Eng. degree from the Department of Electronic Engineering, Dalian University of Technology, in 2002, the M.Phil. degree from the Department of Radio Engineering, Southeast University, in 2005, and the Ph.D. degree from the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, in 2009. He held various research positions with the University of Illinois at Urbana-Champaign, The Hong Kong University of Science and Technology, and McMaster University. He is an Associate Professor with the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, and also the Interim Head of the newly established Centre for Artificial Intelligence and Robotics. His research interests include multimedia security and forensics, multimedia signal processing, artificial intelligence, and big data. He holds four granted U.S. patents and two granted Chinese patents. He has coauthored two papers that received the Best Paper Award at the IEEE Pacific-Rim Conference on Multimedia in 2007 and the Best Student Paper Award at the IEEE International Conference on Multimedia and Expo in 2016. He is serving as the Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING and the IEEE TRANSACTIONS ON MULTIMEDIA.

Jinyu Tian (Student Member, IEEE) received the B.S. and M.S. degrees in mathematics from Chongqing University, Chongqing, China, in 2014 and 2017, respectively. He is currently pursuing the Ph.D. degree with the Faculty of Science and Technology, University of Macau, Macau, China. His research interests include deep learning, subspace learning, and adversarial machine learning.

Jun Liu (Student Member, IEEE) received the B.Eng. and M.S. degrees in software engineering from Beihang University, Beijing, China, in 2015 and 2018, respectively. She is currently pursuing the Ph.D. degree with the Faculty of Science and Technology, University of Macau, Macau, China. Her current research interests include multimedia security and forensics, online social networks, and adversarial machine learning.

Yu Qiao (Senior Member, IEEE) is currently a Professor with the Shenzhen Institute of Advanced Technology (SIAT), Chinese Academy of Sciences, and the Director of the Institute of Advanced Computing and Digital Engineering. He has published more than 180 papers in international journals and conferences, including IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (TPAMI), *IJCV*, IEEE TRANSACTIONS ON IMAGE PROCESSING (TIP), IEEE TRANSACTIONS ON SIGNAL PROCESSING (TSP), *CVPR*, and ICCV. His research interests include computer vision, deep learning, and bioinformation. He received the First Prize of Guangdong Technological Invention Award and the Jiaxi Lv Yong Researcher Award from the Chinese Academy of Sciences. His group achieved the First Runner-Up at the ImageNet Large Scale Visual Recognition Challenge 2015 in scene recognition and the Winner at the ActivityNet Large Scale Activity Recognition Challenge 2016 in video classification. He served as the Program Chair for IEEE ICIST 2014.