Attentive and Contrastive Image Manipulation Localization With Boundary Guidance

Wenxi Liu, *Member, IEEE*, Hao Zhang, Xinyang Lin, Qing Zhang, Qi Li, Xiaoxiang Liu, and Ying Cao

***Abstract*— In recent years, the rapid advancement of image generation techniques has resulted in the widespread abuse of manipulated images, leading to a crisis of trust and affecting social equity. Thus, the goal of our work is to detect and localize tampered regions in images. Many deep learning based approaches have been proposed to address this problem, but they can hardly handle the tampered regions that are manually fine-tuned to blend into image background. By observing that the boundaries of tempered regions are critical to separating tampered and non-tampered parts, we present a novel boundary-guided approach to image manipulation detection, which introduces an inherent bias towards exploiting the boundary information of tampered regions. Our model follows an encoder-decoder architecture, with multi-scale localization mask prediction, and is guided to utilize the prior boundary knowledge through an attention mechanism and contrastive learning. In particular, our model is unique in that 1) we propose a boundary-aware attention module in the network decoder, which predicts the boundary of tampered regions and thus uses it as crucial contextual cues to facilitate the localization; and 2) we propose a multi-scale contrastive learning scheme with a novel boundary-guided sampling strategy, leading to more discriminative localization features. Our state-of-art performance on several public benchmarks demonstrates the superiority of our model over prior works.**

***Index Terms*— Image manipulation detection/localization.**

I. Introduction

IMAGE editing and generation technology [1], [2], [3], [4] has been rapidly developed during the past few years. Latest image editing software and image generation applications allow users to manipulate images freely, which can be easily misused to extort, spread fake news, and even commit crimes. Thus, there are growing concerns on the abuse of manipulated images which may severely disturb the lives of people. As a rising research topic, image manipulation detection, or image tampering detection, is of significance in realistic applications and it has been drawing increasing interests from computer vision community.

Manuscript received 15 August 2023; revised 15 December 2023 and 14 April 2024; accepted 20 June 2024. Date of publication 8 July 2024; date of current version 16 July 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62072110, Grant U21A20471, and Grant U21A20472. The associate editor coordinating the review of this article and approving it for publication was Dr. Weizhi Meng. *(Corresponding author: Ying Cao.)*

Wenxi Liu, Hao Zhang, Qing Zhang, Qi Li, and Xiaoxiang Liu are with the College of Computer and Data Science, Fuzhou University, Fuzhou 350025, China.

Xinyang Lin is with Xiamen Zhonglian Century Company Ltd., Xiamen 361000, China.

Ying Cao is with the School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China (e-mail: caoying59@gmail.com).

Digital Object Identifier 10.1109/TIFS.2024.3424987

The goal of our paper is to localize the image manipulation of different types (including splicing, copy-move, and removal) on pixel level. The major challenge lies in the difficulty of distinguishing between the tampered and non-tampered areas, especially the tampered areas are copied from the original image and their tones are carefully fine-tuned. So, the discrepancy between the tampered and non-tampered areas becomes minor. Prior methods aim to learn the task-specific salient features [6], [7], [8], [9], but either they can only handle certain manipulation types [10], [11], [12], [13], [14], [15], or they will easily be confused by carefully manipulated images.

In a manipulated image, the boundaries of the tampered regions serve as key places to separate manipulated and unmanipulated pixels, which should be paid special attention to and leveraged explicitly when localizing the tampered regions. Nevertheless, how to capitalize on such boundary information to improve the performance of detecting manipulated image regions is still under-explored.

In this work, we propose a boundary-aware scheme for image manipulation detection, where we introduce an inductive bias towards fully exploiting the boundary information of tampered regions and realize our scheme from two perspectives: *attention* and *feature learning*. First, the proposed model is encouraged to attend on the boundary around a tampered region, in which unnatural blending often exists, in order to further enhance manipulation localization. Second, inspired by the recent progress on contrastive learning [16], [17], [18], [19], [20], we seek to learn a feature space, where points within a tampered region are far away from non-tempered region points near the tampered region boundary, in order to obtain more powerful features for localizing tampered regions.

In contrast to prior works that naively exploits boundary information [21], such as jointly predicting tampering boundaries and masks (see Fig. 5), our *attentive* and *contrastive* approach offers a novel, more sophisticated means of leveraging boundary information, and is shown to be more effective than the prior methods.

On the attention side, in the decoding layers of our framework, we propose a novel cross-attention based boundary-aware module that aims to extract the boundary of the tampered area in a image, so that the model will further concentrate on the boundary of the tampered area. In particular, the boundary-aware attention module exploit the correlation of skip-connected encoded features and the decoded features from the previous layer to extract the boundary of the tampered area, which is further used to generate the mask of manipu-

lation localization. On the feature learning side, our proposed model is based on a typical encoder-decoder architecture and its feature learning is supervised by a novel contrastive objective function [16], [22], [23], denoted as boundary-guided tampering contrastive loss, in order to push apart the features sampled from the tampered and non-tampered areas and thus learn more discriminative feature representation. To do so, we adopt a boundary-guided sampling strategy to gather negative training pairs, where we sample negative examples around the boundary of the tampered region instead of the entire non-tampered region. This sampling scheme not only encourages the model to focus on the border region where unnatural blending exists, but also mitigates the distraction caused by the large variation within non-tampered area (see the visualized features in Fig. 1).

For evaluation, we conduct experiments on several public datasets including CASIA [24], Columbia [25], Coverage [26], and NIST16 [27]. By comparing our approach against the previous methods, it is demonstrated that our proposed model can achieve the state-of-the-art performance. In summary, the contributions of our work include:

- We present a novel boundary-guided image manipulation localization model that fully exploits the boundary information of tampered regions via carefully designed attention and contrastive learning mechanisms, as opposed to the naive strategies of using boundary information in previous works.
- We introduce a boundary-aware attention module in the decoder of our framework, which intends to guide the model to emphasize on the unnatural blending of image manipulation by extracting the boundary of the manipulated areas.
- We propose a boundary-guided tampering contrastive loss that encourages the model to enlarge the margin of the samples from the tampered and non-tampered areas to the largest extent.
- We conduct extensive evaluations comparing our method against existing methods on several benchmarks and show that our method achieves state-of-the-art performance.

II. Related Works

In this section, we survey the relevant literatures on image manipulation detection/localization, deepfake detection, and contrastive learning.

A. Image Manipulation Detection/Localization

Image manipulation detection/localization concentrate on the task of identifying the manipulated pixels in images. Previous studies [10], [11], [12], [13], [15], [28], [29], [30], [31], [32], [33] focusing on image manipulation localization, are usually limited to one particular type of manipulation, *e.g.*, copy-move (copy and move elements from one region to another region in a given image), splicing (copy elements from one image and paste them on another image) and removal (removal of unwanted elements). But manipulation operations are usually unknown in practice and combined to apply to the same image, which makes image manipulation detection and localization more challenging. With the success

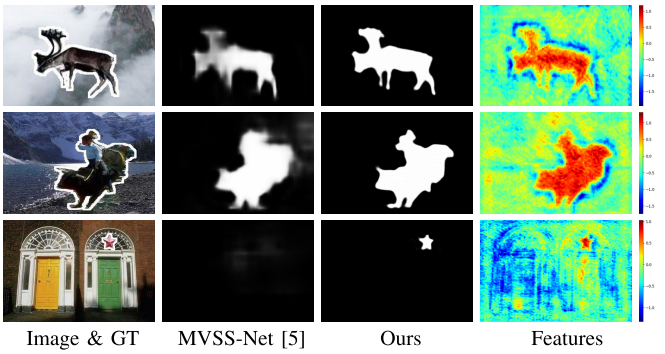

Fig. 1. Our boundary-guided image manipulation detection model is capable of more precisely and robustly localizing tampered image regions with more sharp and complete predicted masks, compared to the state-of-the-art method, MVSS-Net [5]. Such performance improvements largely come from more discriminative and boundary-aligned features (as visualized in the last column) learned through our boundary-aware attention and contrastive learning. We visualize the feature maps, where the pixels with warmer color represent the regions with higher attention and thus higher likelihood of being tampered, and vice versa.

of deep learning techniques in various computer vision tasks, a number of recent attempts have been made to address image manipulation detection [5], [32], [33], [34], [35], [36], [37], [38], [39]. As manipulating a specific image region inevitably leaves traces between the tampered area and its surrounding area, several methods exploit the boundary information to benefit manipulation detection [5], [21], [35]. Multi-Task Fully Convolutional Network (MFCN) [35] proposes to utilize two output branches which localize the boundary of the splicing region. As a GAN-based model, GSR-Net [21] learns to detect image manipulation via synthesizing manipulated images from existing datasets, which is jointly supervised by tampering region and its boundary. Reference [5] presents a two branches network MVSS-Net which fuses the noise distribution and edge information extracted via Sobel to accomplish image manipulation localization. Our method shares a high-level idea of seeking to leverage manipulation boundary information, but explores two novel approaches that make use of contrastive learning and attention mechanism to incorporate the boundary prior, which has not been studied by prior works in the context of image manipulation detection.

B. Deepfake Detection

In recent years, with the development of generative models, the task of deepfake detection has drawn the attention of researchers [40], [41], [42], [43], [44], [45]. It aims to identify images where the expressions and even the identity of human faces are manipulated. Essentially, deepfake detection solves an image-level binary classification problem. Different from the deepfake detection task, our image manipulation localization task requires to estimate the locations of manipulated image regions, which is a pixel-level prediction task.

C. Contrastive Learning

Unsupervised/self-supervised learning methods [20], [22], [46], [47] generally consists of two aspects: pretext tasks and loss functions. Both of them aim to better learn data representations. In recent years, contrastive learning loss has achieved remarkable progress [16], [22], [23], [48]. These

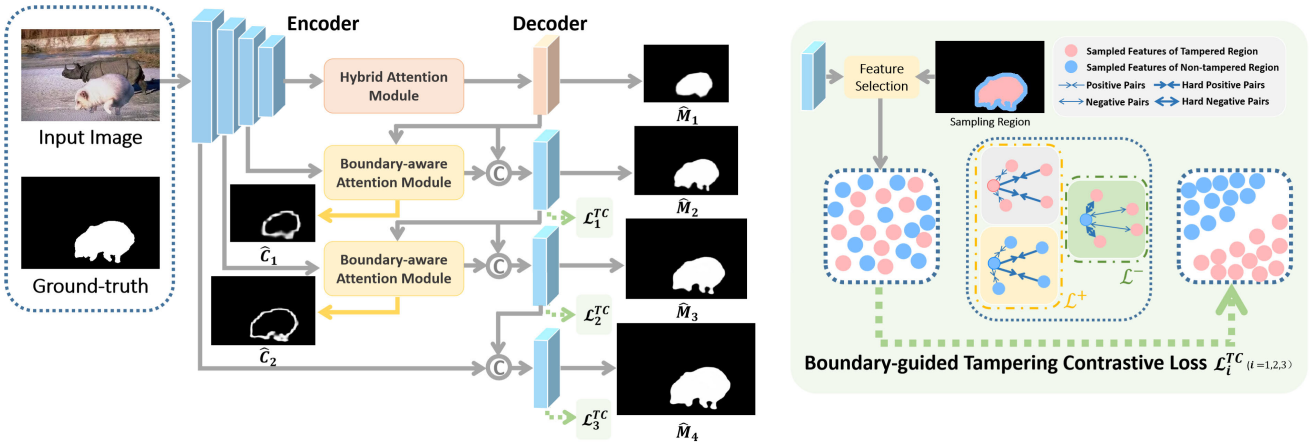

Fig. 2. Left: Overview of our framework. The encoded features of the input manipulated image is passed through a hybrid attention module and several boundary-aware attention modules to predict the multi-scale masks of tampered regions and boundaries. The decoder of each scale is additionally supervised by a boundary-guided tampering contrastive loss. Right: Boundary-guided tampering contrastive loss. Given the feature map of the decoder at a scale, we apply contrastive learning to group point features that belong to the same region (*i.e.*, tampered or non-tampered region), while separating point features from different regions. The sampling of point features from the non-tampered region is restricted to the exterior boundary region of the tampered region and a hard pair mining approach is used to make the model focus on the types of sample pairs that it has difficulty in dealing with - a positive pair with two distant samples (hard positive pair) and a negative pair with two close samples (hard negative pair).

approaches learn representations by pulling close the samples from the positive pair and pushing apart the samples from the negative pair. Reference [49] uses a memory bank to store the instance class representation vector and proposes instance-wise contrastive learning by distinguishing different instances from feature representations. Other works [50], [51] explore to select positive and negative samples in a batch instead of a memory bank. MoCo [23] presents unsupervised visual representation learning, which constructs a dynamic dictionary with a queue and a moving-averaged encoder from the perspective of comparative learning. For image manipulation, a recent work, CFL-Net [52], proposes to use contrast learning to separate the distributions of the untampered and manipulated patch embeddings. By contrast, we propose a boundary-guided sampling strategy to find more informative negative pairs for contrastive learning, enabling our learned features more discriminative.

III. Our Method

A. Network Overview

The goal of our work is to detect and locate the tampered regions on pixel-level. The architecture of our proposed model is illustrated in Fig. 2.

Given a manipulated image I as input, we employ ResNet-50 [53] as the backbone to extract multi-scale visual features, X_i $(i = \{1, \ldots, S\})$. Next, the features are fed into the hybrid attention module that consists of a channel-attention block and a spatial-attention block cascaded successively, in order to transform features and thus locate the potential tampered regions preliminarily before delivering into the decoder. Based on [54], [55], [56], and [57], we employ the hybrid attention module that can model the long-range dependency of the deepest encoded features X_S along the spatial dimension and channels. Concretely, it is composed of a channel-attention block F_{ch}, and a spatial-attention block F_{sp} cascaded successively, which are realized via self-attention schemes along channel and spatial dimension, respectively. After that, X_S will join with the encoded features to obtain $\hat{X}_S$, *i.e.*, $\hat{X}_S = \texttt{Concat}(X_S, F_{ch}(X_S), F_{sp}(F_{ch}(X_S)))$, before passing through the decoder.

In the following decoding layers, the features will be not only upsampled but also interacted with cross-scale features to localize the boundary of the tampered regions via a boundary-aware attention module. The boundary-aware attention module of each scale will estimate the manipulated region mask $\hat{M}_i$ and its boundary $\hat{C}_i$ simultaneously.

On the other hand, to encourage the model to concentrate on the boundary of tampered regions, we propose a boundary-guided tampering contrastive loss based on a boundary-guided sampling strategy in favor of distinguishing tampered and non-tampered region.

In order to provide a more intuitive description of all the symbols used in the paper, we list all the symbols and their connotation in Table I. In the following sections, we will elaborate on our boundary-aware attention module and boundary-guided tampering contrastive loss.

B. Boundary-Aware Attentive Learning

The key to the manipulation detection and localization is the boundary of tampered regions where unnatural blending occurs. Accurate localization of the boundary for tampered regions can effectively assist to locate the tampered regions. In the decoder of our network, we incorporate so-called boundary-aware attention modules F_{ba} that specifically aim to estimate the boundary.

In order to locate the boundary, we not only require the features from the previous layer, but also the features with semantics and details. As the spatial dimension of the decoded features increases, more detailed information is needed. Thus, inspired by UNet-like network structure [58], we utilize the encoded features on the same scale, X_i, along with the decoded features of the previous layer, $\hat{X}_{i+1}$, to facilitate the boundary localization. There are two outputs from

TABLE I
ILLUSTRATION OF SYMBOLS USED IN THE SUBSEQUENT SECTIONS AND THEIR MEANING

Symbol	Explanation
I	Input image
F_{ch}	Channel-attention block
F_{sp}	Spatial-attention block
F_{ba}	Boundary-aware attention module
F_{fs}	Feature selection block
X_i $(i = \{1, ..., S\})$	Multi-scale features derived from the backbone
$\tilde{X}_i$ $(i = \{2, 3\})$	The decoded feature of F_{ba}
$\hat{X}_i$ $(i = \{1, ..., S\})$	The final decoded feature for mask prediction at the i-th scale
$\tilde{H}_i$ $(i = \{2, 3\})$	The combined feature of X_i and $\hat{X}_{i+1}$
H_i $(i = \{2, 3\})$	Decoded feature for boundary prediction
$\mathbf{v}_i$ $(i = \{2, 3\})$	The K most relevant features to the boundary selected from X_i
$\hat{\mathbf{v}}_{i+1}$ $(i = \{2, 3\})$	The K most relevant features to the boundary selected from $\hat{X}_{i+1}$
$\hat{\mathbf{v}}_i$ $(i = \{2, 3\})$	Boundary feature enhanced by cross-attention
$\hat{M}_i$ $(i = \{1, ..., S\})$	Prediction mask at the i-th scale
$\hat{C}_i$ $(i = \{2, 3\})$	Boundary prediction mask at the i-th scale

the boundary-aware attention modules, *i.e.*: 1) the predicted boundary $\hat{C}_i$ and 2) the features that will be propagated to the next layer and used to spawn the mask of the i-th scale, $\hat{M}_i$. The process can be expressed as below.

$$\begin{aligned}[\tilde{X}_i, \hat{C}_i] &= F_{ba}(\hat{X}_{i+1}, X_i),\\ \hat{X}_i &= \texttt{Concat}(\tilde{X}_i, \texttt{Upsample}(\hat{X}_{i+1})),\\ \hat{M}_i &= \texttt{Conv}(\hat{X}_i).\end{aligned} \tag{1}$$

In Fig. 3, inspired by [59], [60], [61], and [62], we design the structure of our boundary-aware attention module. First, we downsample the encoded features X_i to match the dimension of the decoded features from the previous layer $\hat{X}_{i+1}$ and concatenate them. To extract the boundary information of the features, we smooth the features via a combo of average pooling, convolution, and sigmoid operators, and let the original features subtract the smoothed features to obtain the high-frequency information relevant to the tampering boundary. The procedure can be described as below.

$$\begin{aligned}\tilde{H}_i &= \texttt{Concat}(\texttt{Downsample}(X_i), \hat{X}_{i+1}),\\ H_i &= \tilde{H}_i - \tilde{H}_i \odot \texttt{Sigmoid}(\texttt{Conv}(\texttt{AvgPool}(\tilde{H}_i))),\end{aligned} \tag{2}$$

where $\tilde{H}_i$ is the combined features of X_i and $\hat{X}_{i+1}$. $\odot$ denotes the element-wise multiplication. The acquired features H_i are employed to predict the boundary, *i.e.*, $\hat{C}_i = \texttt{Conv}(H_i)$.

Next, the features need to be used for generating the manipulation mask $\hat{M}_i$ and delivering to the next layer, so the most relevant information to tampering should be exploited. Here, we employ a feature selection block F_{fs} that finds the indices of the top-K high confidence pixels from the predicted boundary map $\hat{C}_i$. Then, those indices guide the model to

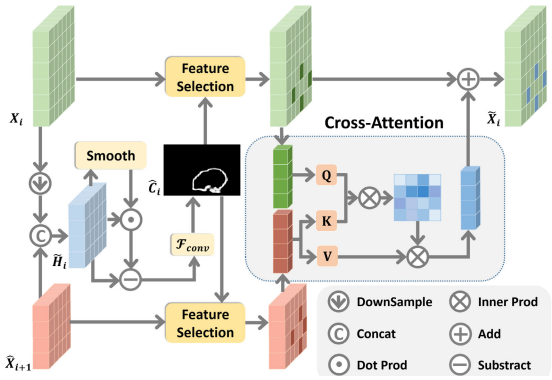

Fig. 3. Illustration of the boundary-aware attention module. The pixel-level features selected from X_i and $\hat{X}_{i+1}$ are denoted by dark green and dark red, respectively. Through the cross-attention module, we obtain the enhanced features $\hat{\mathbf{v}}_i$, indicated in blue, which are then mapped back onto X_i while preserving the remaining features unchanged.

sample the corresponding of pixel-level features of X_i and $\hat{X}_{i+1}$, denoted as $\mathbf{v}_i$ and $\hat{\mathbf{v}}_{i+1}$.

$$\mathbf{v}_i = F_{fs}(X_i, \texttt{TopK}(\hat{C}_i)), \hat{\mathbf{v}}_{i+1} = F_{fs}(\hat{X}_{i+1}, \texttt{TopK}(\hat{C}_i)), \tag{3}$$

where K is set as 32 in practice. Since $\mathbf{v}_i$ and $\hat{\mathbf{v}}_{i+1}$ are the most crucial features for localizing boundary, we employ a cross-attention module with $\mathbf{v}_i$ as the query and $\hat{\mathbf{v}}_{i+1}$ as the key/value as below.

$$\hat{\mathbf{v}}_i = \mathbf{v}_i + \texttt{Softmax}(\frac{\mathbf{v}_i \hat{\mathbf{v}}_{i+1}^T}{\sqrt{d_k}})\hat{\mathbf{v}}_{i+1}, \tag{4}$$

where d_k denotes the dimension of $\mathbf{v}_i$ and $\hat{\mathbf{v}}_{i+1}$. Finally, we scatter the $\hat{\mathbf{v}}_i$ into X_i according to the selected indices and retain the unselected positions unchanged.

Loss Function: We apply the multi-scale loss in order to learn representative multi-scale features for more precise prediction. In practice, as shown in Fig. 2, the manipulation masks are generated for all the four scales, while the boundary masks are only predicted for the intermediate two scales. For the predicted mask by the deepest features, *i.e.*, $\hat{M}_S$, we apply a binary cross-entropy (BCE) loss and an IoU loss for supervision. In addition, we adopt the weighted binary cross-entropy loss [63] and the weighted IoU loss [63] to supervise the predicted masks $\hat{M}_i$ $(i \neq S)$ and boundaries $\hat{C}_i$. The ground truth of the boundary is obtained by subtracting the erosion of the binary ground truth mask of the tampered image from its dilated one. Specifically, we apply `MaxPooling` operation with a kernel size of 5×5 and a stride of 1 to perform image dilation and erosion.

C. Boundary-Guided Tampering Contrastive Learning

Once an image has been manipulated, its tampered areas may exhibit slightly different visual statistics from the non-tampered areas, *e.g.*, unnatural illumination or inconsistent noise distribution. Widening the difference between tampered and non-tampered regions in the learned feature space could effectively improve the discriminative power of the learned features, which is beneficial to tampered region localization.

In light of this, we employ contrastive learning, with the goal of learning discriminative feature representations that

can distinguish between tampered and non-tampered parts. We choose to perform contrastive learning, for each scale, on the feature maps that are directly used to predict the final mask $\hat{M}_i$, which we empirically found works well. In particular, during training, we spatially sample the feature maps in a point-wise manner to collect samples from both the tampered and non-tampered regions. Here, a sample refers to a feature vector at a particular location of the feature maps. Then, we minimize a contrastive loss function to reduce the distance between samples within the same region (*i.e.*, positive pair) while increasing the distance between samples in different regions (*i.e.*, negative pair). To further improve the robustness and discriminativeness of learned features, we introduce a boundary-guided sampling strategy to construct more informative negative pairs.

1) Boundary-Guided Tampering Contrastive Loss: From each scale of the decoder, we employ a contrastive loss, which consists of two terms, *i.e.*, $\mathcal{L}^{TC} = \mathcal{L}^{+} + \mathcal{L}^{-}$, where $\mathcal{L}^{+}$ and $\mathcal{L}^{-}$ are a positive pair loss and a negative pair loss, respectively. Formally, $\mathcal{L}^{+}$ is written as:

$$\mathcal{L}^{+} = \frac{1}{|\mathcal{H}_t^{+}|} \sum_{(\mathbf{u}_m, \mathbf{u}_n) \in \mathcal{H}_t^{+}} -\log(\texttt{Sim}(\mathbf{u}_m, \mathbf{u}_n)) + \frac{1}{|\mathcal{H}_n^{+}|} \sum_{(\mathbf{u}_m, \mathbf{u}_n) \in \mathcal{H}_n^{+}} -\log(\texttt{Sim}(\mathbf{u}_m, \mathbf{u}_n)), \quad (5)$$

where $\texttt{Sim}(\cdot, \cdot)$ denotes cosine distance between two feature vectors and $\mathcal{H}_t^{+}$ is a set of positive pairs from the tampered region, while $\mathcal{H}_n^{+}$ contain positive pairs drawn from the non-tampered region. $(\mathbf{u}_m, \mathbf{u}_n)$ represent the sampled pair of feature vectors. $\mathcal{L}^{+}$ serves as a force to pull together the samples within the same region in the feature space. To obtain the positive pairs of samples, we use a hard pair mining strategy to bring together the samples that are within the same region but far away from each other. Specifically, among all possible sample pairs from the tampered or non-tampered region, we retain the sample pairs with the top-L largest distances to constitute $\mathcal{H}_t^{+}$ and $\mathcal{H}_n^{+}$. In practice, L is set as the half of the number of all possible sample pairs from the same region. We define $\mathcal{L}^{-}$ as below:

$$\mathcal{L}^{-} = \frac{1}{|\mathcal{H}^{-}|} \sum_{(\mathbf{u}_m, \mathbf{u}_n) \in \mathcal{H}^{-}} -\log(1 - \texttt{Sim}(\mathbf{u}_m, \mathbf{u}_n)), \quad (6)$$

where $\mathcal{H}^{-}$ is a negative pair set, where each pair is comprised of a query sample from the tampered region and a negative sample from the non-tampered region. $\mathcal{L}^{-}$ aims to push apart samples from different regions.

2) Boundary-Guided Sampling Strategy: To construct $\mathcal{H}^{-}$, one naive approach is to sample negative samples at random from a non-tampered region. However, there usually exists large variance within a non-tampered region. For example, the non-tampered region may occupy a large proportion of the image, so it may contain plenty of distraction from various objects or cluttered background. Hence, the native way of drawing negative samples tends to degrade the model performance.

Therefore, we propose a novel sampling strategy, in which we draw negative samples near the boundary of a tampered region, rather than from the entire non-tempered region (see the example in Fig. 4). Such sampling approach is advantageous in two aspects: first, constraining negative samples within the boundary area tends to reduce the distraction caused by the large variance within the non-tampered region; second, unnatural blending usually occurs around the boundary of tampered regions, so sampling from this area encourages the model to focus on the boundary regions, which fits with the other components of our framework, *e.g.*, the boundary-aware attention modules. In particular, given a ground-truth binary tampered region mask M, we perform image dilation on it to get a dilated mask $\tilde{M}_i$ and restrict negative samples to be from the exterior boundary region specified by $\tilde{M}_i - M_i$. Finally, we take a hard pair mining approach as for $\mathcal{H}_t^{+}$ and $\mathcal{H}_n^{+}$, only retaining the negative pairs with the top-L smallest distances to construct $\mathcal{H}^{-}$.

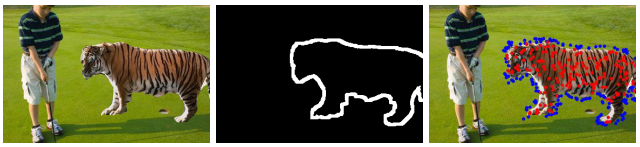

Fig. 4. Boundary-guided sampling strategy. The blue points represent sampling features in non-tampered region and the red ones represent sampling features in tampered region.

IV. Experimental Results

In this section, we conduct comprehensive experiments on several public benchmarks and compare our proposed method against previous state-of-the-art methods. We also analyze different components of our model.

A. Implementation Details and Datasets

Implementation details We implement our framework using Pytorch on the workstation with a single NVIDIA Titanxp GPU for both training and testing. For training, all the input images are resized to a resolution of 512×512 and they are augmented by randomly horizontal flipping, color jittering, and cropping. We adopt ResNet50 [53] as our backbone. During training, we use the Adam optimizer [64] with a momentum of 0.9 and a weight decay of 5×10^{-4}. We set the batch size to 16 and adjust the learning rate with the polynomial strategy [65] with a basic learning rate of 1×10^{-5} and the power of 0.9. For testing, the input image is first resized to 512×512 for network inference and then the output map is resized back to the original size of the input image.

Datasets To evaluate the performance of the model, we experiment on six benchmarks including CASIA [24], NIST16 [27], Columbia [25], Coverage [26], Defacto [67], and a real-world dataset IMD2020 [68]. In addition, we also utilize the synthetic dataset proposed from [69]. The datasets cover different types of manipulation, including splicing, copy-move and removal. All datasets provide the ground-truth binary masks.

- **CASIA** [24] consists of two sub-datasets, CASIAv1 with 921 tampered images and CASIAv2 with 5123 tampered images. The manipulation types include splicing

TABLE II
COMPARISON OF DIFFERENT METHOD IN TERMS OF F1 (%) WITH A FIXED THRESHOLD OF 0.5 ON THE FOUR BENCHMARKS. ALL THE MODELS ARE TRAINED ON CASIAV2 AND TESTED ON THE BENCHMARKS. THE RESULTS OF THE OTHER METHODS ARE TAKEN FROM [5]

Method	CASIAv1	NIST16	Columbia	Coverage	IMD20	DEFACTO	MEAN
ManTra-Net [37]	15.5	0.0	36.4	28.6	18.7	**15.5**	19.1
HP-FCN [71]	15.4	12.1	6.7	0.3	11.2	5.5	8.5
CR-CNN [72]	40.5	23.8	43.6	29.1	26.2	13.2	29.4
GSR-Net [21]	38.7	28.3	61.3	28.5	24.3	5.1	31.0
SPAN [66]	18.4	22.1	48.7	17.2	17.0	4.8	21.4
CAT-Net [73]	13.6	17.9	55.5	12.9	5.4	4.6	18.3
MVSS-Net [5]	45.2	29.2	63.8	45.3	26.0	13.7	37.2
MVSS-Net++ [74]	51.3	30.4	66.0	48.2	27.0	9.5	38.7
WSCL [75]	15.3	9.9	36.2	20.1	17.3	-	-
Ours	**60.0**	**33.3**	**67.9**	**49.0**	**39.7**	12.3	**43.7**

and copy-move. The cropped regions are often carefully selected and post-processing operations are applied to make the regions visually realistic.

- **NIST16** [27] contains 564 tampered image samples. All three manipulation types are involved and they are also post-processed to hide visible traces.
- **Columbia** [25] focuses on splicing and it contains 180 uncompressed images.
- **Coverage** [26] focuses on copy-move manipulation which contains 100 images. The manipulated objects are manually cropped to cover similar objects in the same image and they are post-processed to remove the visible traces of manipulation.
- **DEFACTO** [67] is a recently proposed large-scale synthetic dataset, containing 149k images that are sampled from MS-COCO [70] and auto-manipulated by copy-move, splicing, and inpainting.
- **IMD2020** [68] contains 2010 real-life manipulated images collected from Internet, which involves all the three manipulations types.
- **Synthetic training dataset** [69] contains around 100k images originally collected from [70], covering the manipulation types of splicing, copy-move, and removal.

In the qualitative results of the following subsections, the feature maps are shown in the color map with warmer color indicating more attention and vice versa. The predicted tampering mask of the comparison methods are visualized in gray-scale images ranging from 0 to 1. Each pixel of the predicted masks implies the certainty of being tampered.

B. Comparison With the State-of-the-Art

Pixel-level Manipulation Detection Following [5] and [34], we compare our proposed method with several state-of-the-art methods under *two experimental settings*.

In the first setting, following MVSS-Net [5], all the compared models are trained from scratch on real data without any extra synthetic datasets. We adopt CASIAv2 [24] for training and test on other public benchmarks including CASIAv1, NIST16, Columbia, Coverage, IMD20 and DEFACTO. We compare our model against several prior methods including ManTra-Net [37], HP-FCN [71], CR-CNN [72], GSR-Net [21], SPAN [66], CAT-Net [73], MVSS-Net [5] and MVSS-Net++ [74]. We do not compare with ObjectFormer [34] since its code is not publicly available. Table II shows the results measured in terms of F1 with fixed threshold 0.5 reported in [5]. We can observe our model achieves the best localization performance on all the datasets. We note that GSR-Net also leverages boundary information in their network by predicting tampering boundaries and masks. Our noticeable improvements from GSR-Net on all the benchmarks suggests the advantage of our proposed attentive and contrastive mechanisms over the simple boundary and mask prediction in terms of exploiting boundary information.

TABLE III
COMPARISON OF DIFFERENT METHODS IN TERMS OF F1 (%) WITH THE OPTIMAL THRESHOLD ON THE THREE BENCHMARKS. ALL THE MODELS ARE PRE-TRAINED ON THEIR OWN SYNTHETIC DATASETS OF DIFFERENT SIZES, FINETUNED AND TESTED ON THE TRAINING AND TESTING SETS OF EACH BENCHMARK. THE RESULTS OF THE OTHER METHODS ARE TAKEN FROM [34]

Method	Synthetic Data Size	CASIAv1	NIST16	Coverage
RGB-N [38]	47k	40.8	72.2	43.7
SPAN [66]	96k	38.2	58.2	55.8
PSCC-Net [69]	100k	55.4	81.9	72.3
ObjectFormer [34]	62k	57.9	82.4	75.8
HiFi-Net [77]	1710k	61.6	85.0	**80.1**
ERMPC [76]	60k	58.6	83.6	77.3
Ours	60k	**66.9**	**93.3**	71.1

In the second setting, following [34], [69], and [76] we first pre-train each model on a synthetic dataset, and then finetune it on the training set of a benchmark before testing it on the benchmark's test set. The compared methods including RGB-N [38], SPAN [66], PSCC-Net [69], ObjectFormer [34], HiFi-Net [77] and ERMPC [76], are trained on their own respective synthetic datasets, fine-tuned on the training set of each benchmark and tested on the test set of the corresponding benchmark. We evaluate all the methods on CASIA, NIST16 and Coverage. For CASIA, 5123 images from CASIAv2 are utilized for fine-tuning and 921 images from CASIAv1 for testing. For NIST16 with 564 images involving all the three manipulations, 404 images are used for fine-tuning and 160 images are used for testing. Coverage with 100 images is divided into 75/25 for fine-tuning and testing.

For this setting, as reported in [34], different methods use their own synthetic datasets with different contents and varying

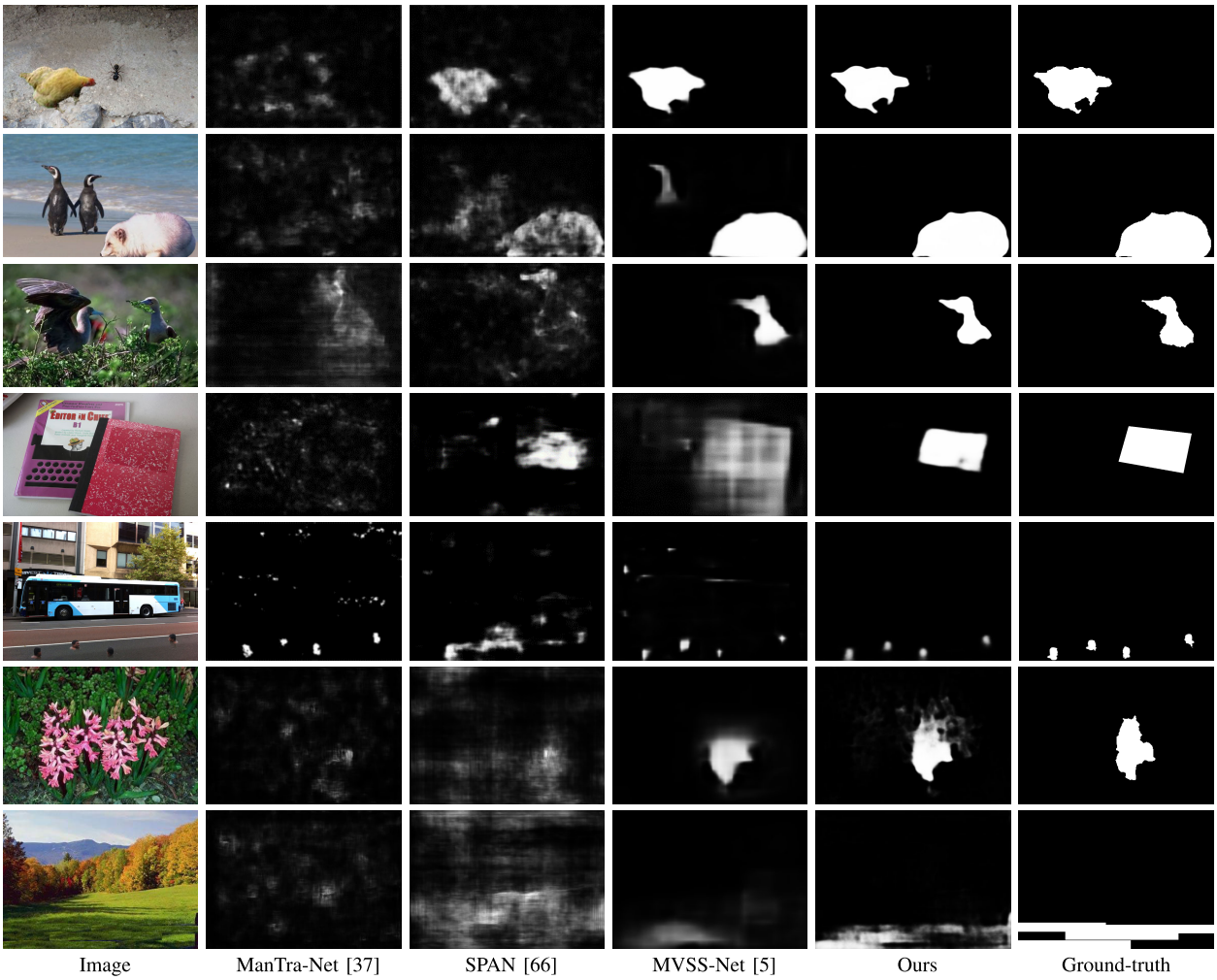

Fig. 5. Comparison of the masks predicted by our proposed method, ManTra-Net [37], SPAN [66], and MVSS-Net [5].

numbers of images ranging from 47k to 100k. In view of this, for fair comparison, we randomly select 60k images from the synthetic dataset of [69]. The results on three benchmarks are shown in Table III. Note that the Columbia has no training set, and thus finetuning on it is not possible. Our method outperforms all the compared methods by a large margin on CASIAv1 and NIST16, even though our method uses significantly less data compared to SPAN and PSCC-Net. On Coverage, our method underperforms some of the lastest methods. Since the test set of Coverage only contains 25 images, it may not be sufficient to accurately reflect the performance of the models. By delving into the failure cases on Coverage (Fig. 12), we find that our model tends to fail on a particular type of images where the manipulation traces around region boundaries are carefully erased. For all of the failure cases shown in Fig. 12, the F1 score of our model is quite low (less than 35%). By excluding these three images from testset, F1 score of our approach goes up to 78.1%.

Fig. 5 shows the qualitative comparison for our method against the state-of-the-art methods. The results demonstrate that our method can not only locate the tampered regions more accurately, but also produce more sharp boundaries (the top three rows of Fig. 5), which benefits from the boundary-aware attention module. Moreover, the results imply that our model is more robust to background distraction than others (the last three rows of Fig. 5), owing to the introduction of contrastive losses for suppressing noises.

Image-level Manipulation Detection While our model focuses more on the pixel-level manipulation localization task, it has the capability of detecting manipulation at image level. The aim of image-level manipulation detection is to classify an input image as authentic or tampered. We follow the protocol of [5] to run an image-level manipulation detection experiment where our model is trained on CASIAv2. Considering that CASIAv1 and CASIAv2 share 782 authentic images, we randomly sample 782 authentic images from the Corel [78] to replace these duplicates in CASIAv1, leading to the dataset CASIAv1+. We show the results in Table IV on three benchmarks, CASIAv1+, Coverage, and Columbia. In this experiment, we compare our model against ManTra-Net [37], CR-CNN [72], GSR-Net [21], SPAN [66] and MVSS-Net [5].

To perform image-level manipulation detection, we modify our model by adding an image classification head to the features $\hat{X}_S$ for predicting the probability of the image being manipulated. Specifically, the image classification head

TABLE IV
IMAGE-LEVEL MANIPULATION DETECTION RESULTS IN TERMS OF F1 (%) WITH FIXED THRESHOLD 0.5 AND AUC (%). THE RESULTS OF PRIOR METHODS WERE REPORTED IN [5]

Method	CASIAv1+		Coverage		Columbia	
	F1	AUC	F1	AUC	F1	AUC
ManTra-Net [37]	0.0	50.0	0.0	50.0	0.0	70.1
CR-CNN [72]	24.2	71.9	13.1	56.6	39.2	78.3
GSR-Net [21]	0.0	50.0	0.0	51.5	2.2	50.2
SPAN [66]	0.0	50.0	0.0	50.0	0.0	50.0
MVSS-Net [5]	75.8	**93.7**	24.4	**73.1**	80.2	**98.0**
Ours	**78.0**	88.1	**54.7**	68.3	**81.3**	94.4

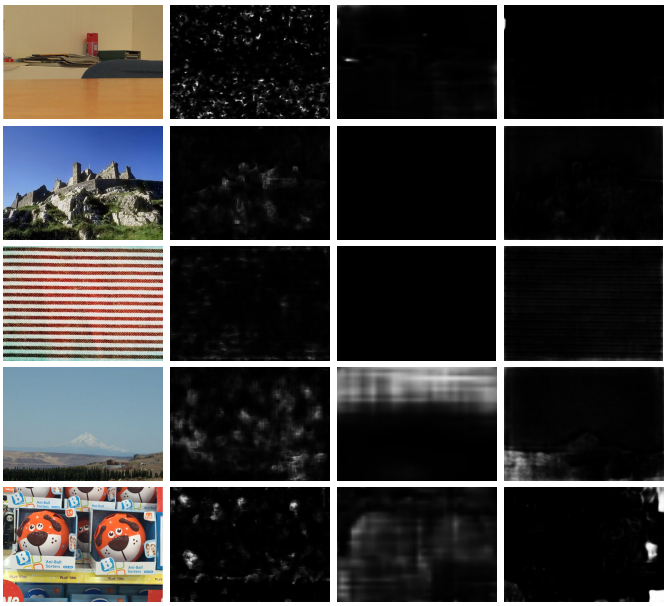

Fig. 6. Qualitative results on untampered images. Note that the ground truth tampering masks of these image are blank. .

consists of a CBR block, an average pooling layer, and a fully connected layer, where the CBR block is the combination of convolution, batch normalization (BN) and ReLU. As observed in Table IV, our approach achieves the best F1 score on all the benchmarks and our model's AUC score is the second best among all the methods, demonstrating that our model is able to yield promising performance for both pixel-level localization and image-level detection tasks.

C. Localization on Untampered Images

We also test our model on untampered images for pixel-level localization, and show the qualitative results in Fig. 6. The predicted masks for the untampered images by our approach are nearly blank on the top three rows of Fig. 6, whereas the compared methods misdetect some positions as tampered. For the last two rows of Fig. 6, all the methods exhibit some false positives. However, our model is prone to restricting the false positives to small local regions while the other methods tend to spread them over the entire image. Furthermore, from the last row of Fig.6, we observe that our model may misdetect suspicious regions in an authentic image as tampered with high confidence. In contrast, MVSS-Net [5] tends to assign lower confidence to the misdetection. This will cause our model to have a higher chance of misclassifying an authentic image as tampered, which partially explains why our model has a lower AUC score comapred to MVSS-Net in the image-level manipulation detection experiment in Section B.

TABLE V
THE EFFECT OF DEPLOYING BAM TO THE DECODING LAYER OF DIFFERENT SCALES

BAM @ i-th Scale			F1 (%)	AUC (%)
3	2	1		
×	×	×	53.9	87.4
✓	×	×	56.9	87.1
✓	✓	×	**60.0**	**88.6**
✓	✓	✓	57.3	86.7

D. Model Component Analysis

To elucidate the effects of individual components, we assess the proposed model in various configurations. All reported results are obtained from the CASIA dataset.

Boundary-aware Attention Module The boundary-aware attention modules can be deployed on each scale of the decoder to predict the boundary, so we verify the design in Table V. For reference, we have a baseline model without any BAMs and it can achieve 53.9% F1 and 87.4% AUC. First, we deploy BAM on the deepest scale (*i.e.*, $i = 3$), which intends to transform the smallest resolution to boundary. As observed, adding a single BAM leads to better performance, which is implied by the gained 3% F1 score. Second, we append another BAM at the second scale (*i.e.*, $i = 2$), so there are two collaborative BAMs for predicting the boundaries of different scales. With more detailed information brought by the encoded features, the model is able to achieve the optimal performance with 60.0% F1 and 88.6% AUC. Last, as we incorporate three BAMs for all scales, the performance obviously drops, because the lowest scale features introduce noises that negatively affect the boundary prediction.

To demonstrate how our boundary-aware attention module can assist in locating tampered regions by detecting boundaries, we present a visualization of feature maps in Fig. 7. To provide a comparison, we also visualize the features produced by the last Sobel layer in MVSS-Net [5], which extracts edge related features. The results in Fig. 7 show that our approach can more accurately locate tampering regions, due to our boundary-aware attention module.

In addition, we show some examples on the effectiveness of BAM in Fig. 8. As observed, without the aid of BAM, the estimated masks tend to be incomplete in the boundary area, since the pixels blend into the background well and they are hard to distinguish. Although the model with BAM may not perfectly predict the boundary, it is sufficient to provide contextual cues for inferring the whole tampered mask.

As shown in Fig. 9, we also show the qualitative results on different datasets comparing to MVSS-Net [5]. To validate the effectiveness of the proposed boundary-aware attention module, we visualize the final masks and predicted boundary

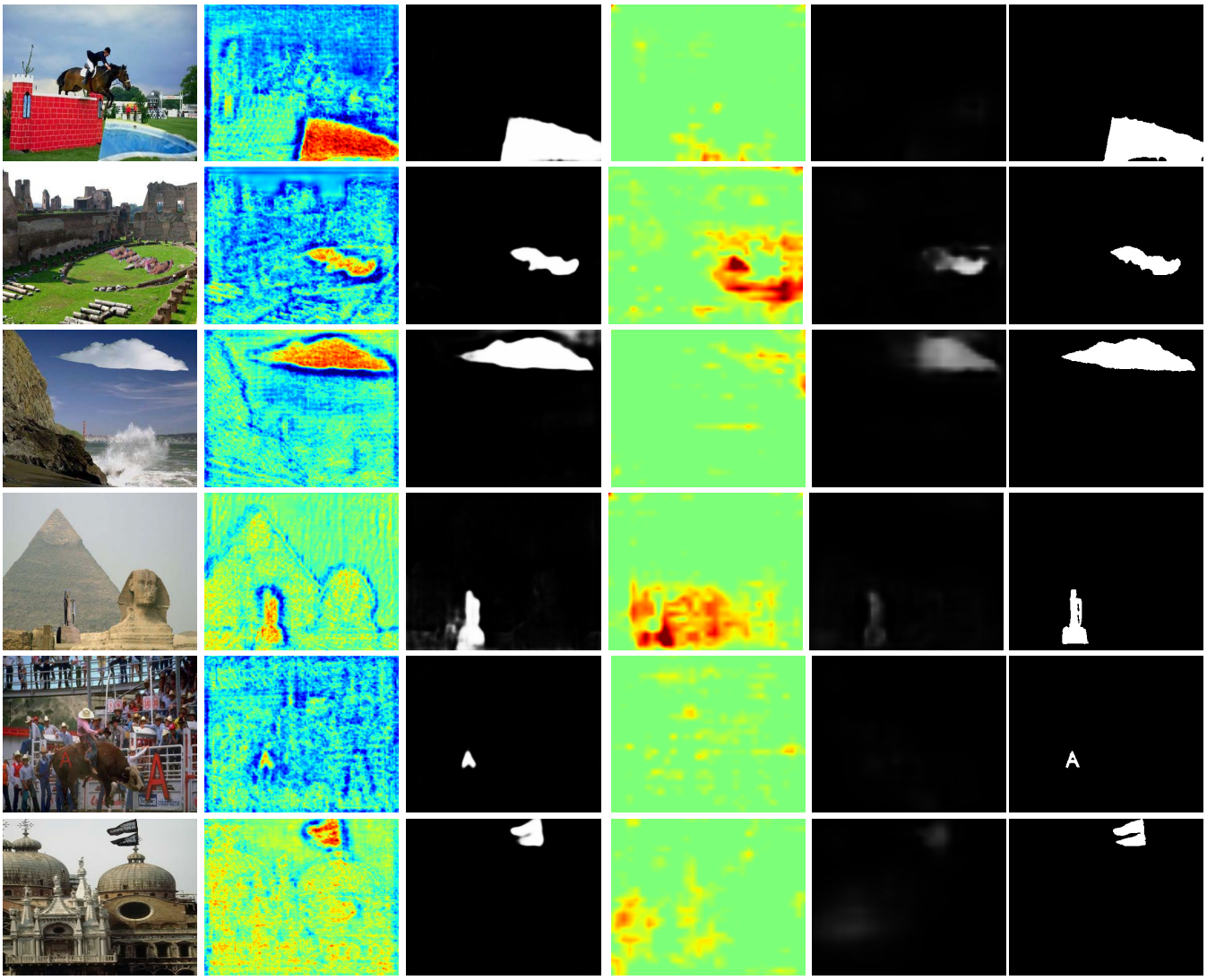

Fig. 7. Comparison of the masks and visualized features predicted by our proposed method and MVSS-Net [5].

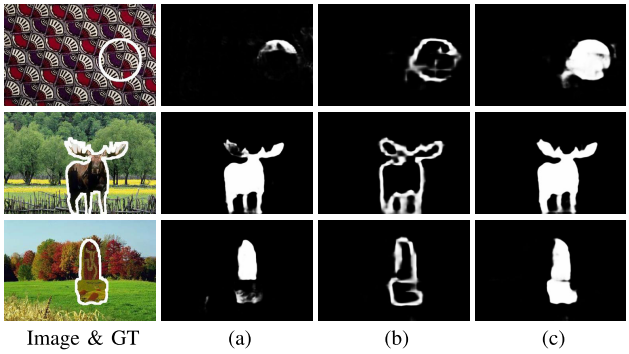

Fig. 8. The effectiveness of BAM. (a) Estimated tampered mask without the aid of BAM. (b) Estimated boundary with the aid of BAM. (c) Estimated tampered mask with the aid of BAM.

masks in the third and fourth columns. As observed in the first two rows, although MVSS-Net can roughly predict the tampered area, its predictions contain large gray areas, indicating lower certainty. In contrast, our prediction results not only can accurately locate the tampered region with sharper boundaries but also exhibit higher certainty for the tampered region. Meanwhile, in the fourth and fifth rows, MVSS-Net may have false negative of the authentic region with large areas of gray, whereas in our predictions, the background areas are nearly black. The results manifest the superiority of our proposed model.

Boundary-guided Tampering Contrastive Loss Recall that the goal of our boundary-guided tampering contrastive loss is to draw the distance of the feature embeddings from the same regions closer while pushing apart the distance of the features from different regions, *i.e.*, tampered and non-tampered regions, so as to make them more discriminative. First, we evaluate the difference on how to deploy the loss in Table VI. As observed, when we employ the losses on all the scales, it generally achieves the optimal results. In contrast, employing the losses on the second and third scales (without the top scale) lead to slightly better AUC (88.9%) but a decrease in F1 (51.5%). This is because the top scale features

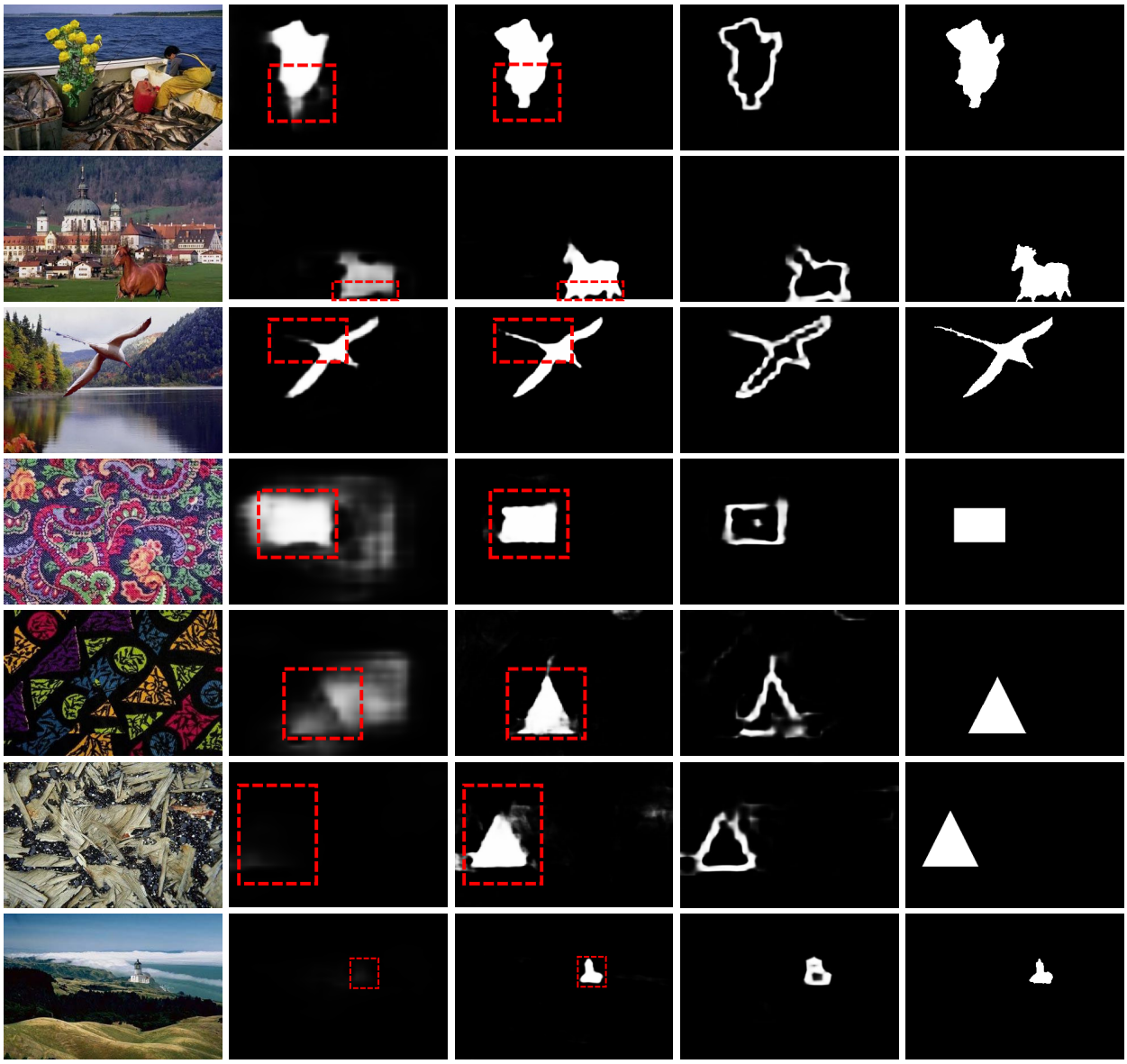

Fig. 9. Comparison of the masks and boundaries predicted by our proposed method and MVSS-Net [5]. We highlight the boundaries of the tampered areas that are mispredicted by MVSS-Net but well distinguished by ours.

TABLE VI

THE EFFECT OF APPLYING THE CONTRASTIVE LOSS ON THE DECODING LAYERS OF DIFFERENT SCALES

Contrastive Loss			F1 (%)	AUC (%)
$\mathcal{L}_3^{TC}$	$\mathcal{L}_2^{TC}$	$\mathcal{L}_1^{TC}$		
×	×	×	50.5	86.9
✓	×	×	52.7	88.3
✓	✓	×	51.5	**88.9**
✓	✓	✓	**60.0**	88.6

TABLE VII

EFFECTIVENESS OF THE BOUNDARY-GUIDED SAMPLING STRATEGY. HSM REPRESENTS HARD PAIR MINING

Sampling Range		Strategy	F1 (%)	AUC (%)
Tampered Area	Non-tampered Area			
All	All	Random	57.6	87.0
All	All	HPM	46.4	88.1
All	Boundary	Random	59.7	85.3
All	Boundary	HPM	**60.0**	**88.6**

directly corresponds to the final results and thus it is related to the manipulation detection accuracy. Without utilizing the top scale features, the general performance becomes a bit worse.

Fig. 10 shows boundary-guided tampering contrastive loss is effective for the image where the tampered area is not obvious. As observed, the discrepancy on the boundary of the tampered region becomes large after using contrastive learning loss. Besides, the tampered regions of the texture images are incomplete parts of certain objects. The results show that our model generalizes well to part-level manipulation.

In addition, Fig. 11 presents the feature maps from different scales after applying the multi-scale contrastive learning loss.

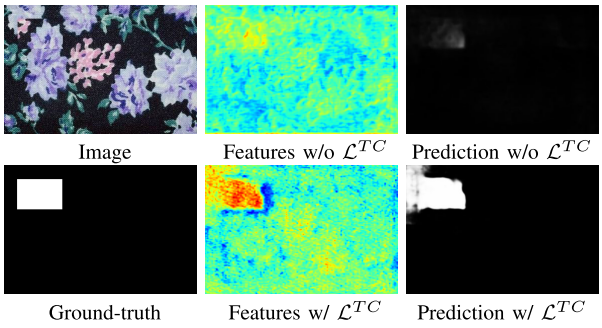

Fig. 10. Efficacy of boundary-guided tampering contrastive loss.

As observed, with the increasing scale in the decoder, the contrastive learning loss localizes the tampered areas in a coarse-to-fine fashion. In the finest scale, we can clearly observe that the color of the boundary around the tampered areas becomes blue while the color of the tampered areas is red, and their large discrepancy is caused by the contrastive learning loss. This phenomenon is not obvious for the features at the 2nd and 3rd scales, so the 1st-scale features provide the most gain for the model improvement.

Hybrid Attention Module We also conduct an ablation study on our hybrid attention module, and find that removing this module results in 0.4% and 2.8% reduction in F1 and AUC. This suggests the importance of this module.

Boundary-guided Sampling Strategy Sampling strategy plays an important role in our proposed contrastive loss. To validate the effectiveness of the sampling strategy, we conduct experiments on the CASIA dataset in Table VII. The most straightforward way is to randomly sample feature points from the tampered area and non-tampered area. In contrast, we employ the hard pair mining for the randomly drawn points from the tampered area and non-tampered area, which increases the performance. However, there exists large variation within the non-tampered area, leading to the unrobustness of contrastive learning results for random sampling from non-tampered area. Thus, we constrain the sampling range within the non-tampered area near the tampering boundary. As observed, randomly selecting points from tampered and non-tampered area near the tampering boundary may bring in large variation among negative samples, resulting in network performance degradation. Finally, we combine the hard pair mining strategy with the samples randomly drawn from boundary area and acquire the optimal performance.

E. Analysis on Hyper-Parameters

We analyze the hyper-parameters of the boundary-guided tampering contrastive loss and boundary-aware attention module on CASIA dataset in terms of F1 and AUC, as depicted in Table VIII. Six critical hyper-parameters are involved.

1) Input Image Size: The training dataset has images of varying sizes, where most of the images have a resolution of 384 × 256, and others have different resolutions such as 640×480, 336×638, 500×375. Thus, we follow the approach of [5] to resize the input image to 512 × 512. We experiment with different input image sizes and find our choice of 512 × 512 gives the best results, as shown in Table VIII.

TABLE VIII
HYPER-PARAMETERS ANALYSIS

Hyper-Parameters		Metrics	
		F1	AUC
Input image size	384×384	57.1	86.2
	512×512	**60.0**	**88.6**
	640×640	59.0	87.6
# of selected indices K	16	57.6	86.7
	32	**60.0**	**88.6**
	64	58.5	87.7
# of dilation/erosion for GT boundary	1	**60.0**	**88.6**
	2	57.9	86.6
	3	57.8	86.2
# of dilation for untampered sampling regions	1	56.0	85.3
	2	58.0	86.7
	3	**60.0**	**88.6**
	4	58.3	86.8
# of samples $\mathcal{Z}$	250	56.6	85.4
	500	**60.0**	**88.6**
Top-L hardest pairs	1	58.3	85.6
	Half	**60.0**	**88.6**
	All	58.1	85.7

2) Number of Selected Indices K: As the key of our proposed boundary-aware attention module, the selected indices correspond to the most relevant features to boundary prediction. We denote the number of the selected indices as K, and set it to be 16, 32, and 64. As depicted in Table VIII, we obtain the best performance when K is 32.

3) Number of Dilation and Erosion for Generating Ground Truth Boundary: Generating a refined ground truth boundary is necessary for supervising the predicted boundary mask $\hat{C}_i$ of the boundary-aware attention module in Section III-B. The process involves deriving the ground truth boundary by subtracting the erosion of the ground truth mask from its dilation. We try to increase the number of dilation and erosion from 1 to 3, which results in increasingly wider ground truth boundaries, and find that narrower ground truth boundaries give better results.

4) Number of Dilation for Untampered Sampling Regions: To perform the boundary-guided sampling strategy, we subtract the ground truth mask of the tampered area from its dilation to obtain the untampered boundary region around the tampered area, from which point features for the untampered region are sampled. We experiment with different numbers of diliation, and find performing dilation for 3 times gives the best performance.

5) Number of Samples $\mathcal{Z}$: For the boundary-guided sampling strategy, after we determine sampling areas for positive and negative sample pairs, we randomly select some of them to construct positive pairs set and negative pair set. To obtain these sample pairs, we need to draw a certain number of samples denoted as $\mathcal{Z}$ from the tampered and non-tampered areas, respectively. Empirically, we set $\mathcal{Z}$ to be 250 and 500. For the tampered area containing less than $\mathcal{Z}$ pixels, we sample the features for all the pixels. We show the results in Table VIII. As observed, a small $\mathcal{Z}$ implies we under-sample both areas and thus may not fully exploit the samples.

6) Top-L Hardest Pairs: For the boundary-guided sampling strategy, we take a hard pair mining approach to obtain the

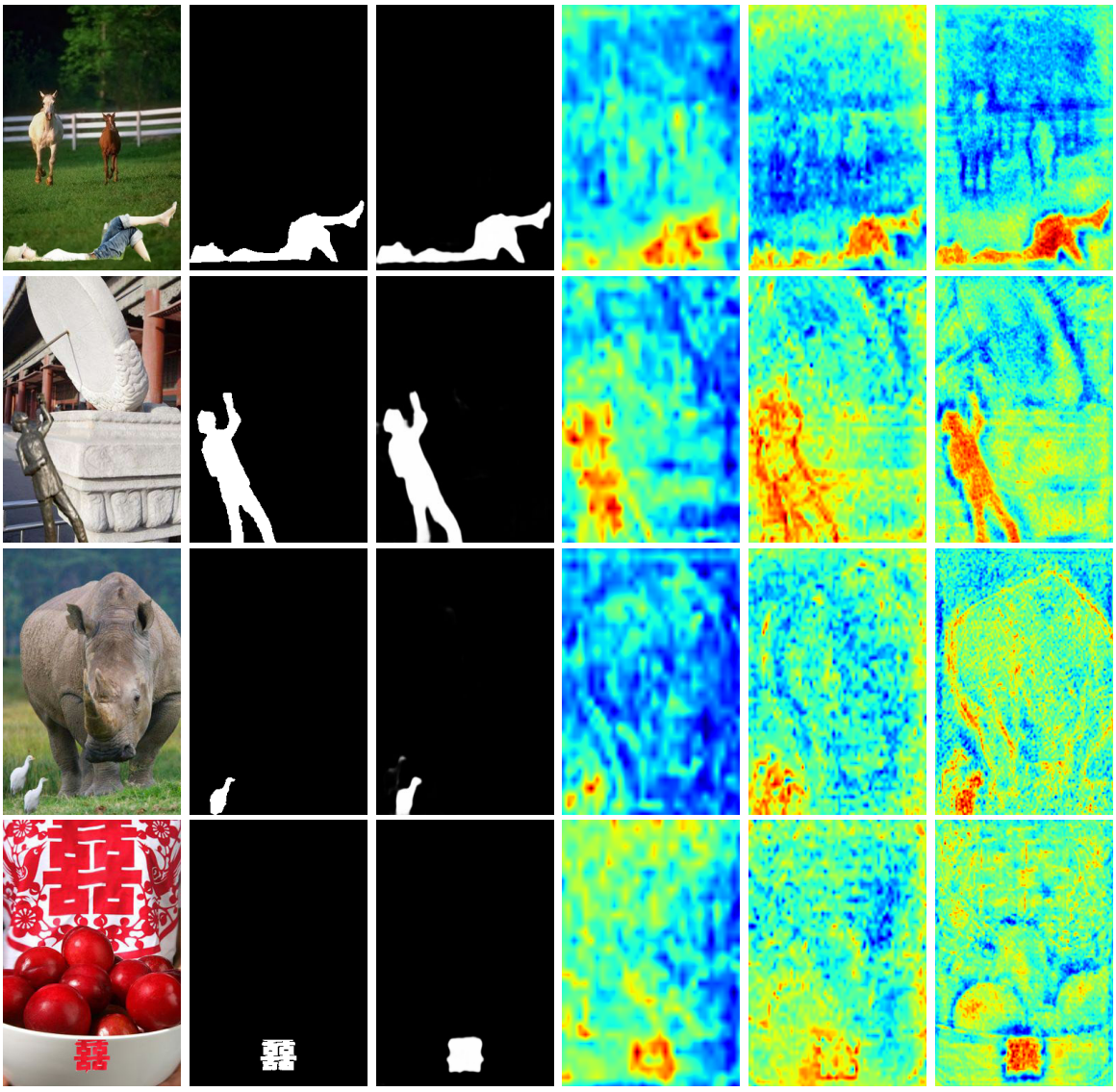

Fig. 11. Visualization of different scales of feature maps under tampering-guided contrastive learning.

TABLE IX

ROBUSTNESS EVALUATION OF PIXEL-LEVEL LOCALIZATION AGAINST VARIOUS DISTORTIONS. PIXEL-LEVEL F1 AND AUC ARE REPORTED

Distortion	Pixel-level Localization			
	MVSS-Net		Ours	
	F1	AUC	F1	AUC
w/o Distortion	45.1	84.5	**60.0**	**88.6**
Resize (0.78×)	44.0	82.7	**51.9**	**84.6**
Resize (0.25×)	11.8	65.4	**26.7**	**70.4**
Gaussian Blur (k=3)	**41.5**	**82.0**	38.1	80.1
Gaussian Blur (k=15)	0.4	51.1	**21.4**	**66.9**
Gaussian Noise (σ=3)	0.0	50.0	**55.2**	**86.9**
Gaussian Noise (σ=15)	0.0	50.0	**41.9**	**81.5**
JPEG Compression (q=100)	23.5	71.2	**50.5**	**84.6**
JPEG Compression (q=50)	7.5	63.0	**19.4**	**68.0**
Resize + Gaussian Blur	16.1	66.6	**28.7**	**73.4**
Resize + Gaussian Noise	0.0	50.0	**41.4**	**78.5**
Resize + JPEG Compression	20.1	70.7	**32.4**	**73.8**
Gaussian Blur + Gaussian Noise	0.0	50.0	**17.8**	**68.2**
Gaussian Blur + JPEG Compression	12.1	66.4	**23.9**	**70.7**
Gaussian Noise + JPEG Compression	0.0	50.0	**26.5**	**74.3**
Mixed	5.4	57.9	**21.5**	**69.4**

positive and negative sample pairs, where we sample top-L hardest pairs and set L to be half of the number of all pairs throughout our experiments. We experiment with L = 1 and L = the number of all pairs. As observed in Table VIII, our top-half strategy achieves the best performance. The top-1 strategy may easily be misled by extreme sample pairs, while the all-pair strategy may suffer from large variation among samples, thus introducing irrelevant information for separating tampered and non-tampered areas.

F. Robustness Evaluation

Following the distortion settings in [5], [34], and [69], we conduct robustness analysis of our network for manipulation localization. Specifically, different distortion operations are applied to manipulated images in CASIA dataset. In addition, we combine any two of the distortions, where the resizing scale, kernel size, standard deviation, and quality factor are randomly selected from the intervals $[0.25, 0.78]$, $[3, 15]$, $[3, 15]$, and $[50, 100]$, respectively.

TABLE X
ROBUSTNESS EVALUATION OF IMAGE-LEVEL DETECTION AGAINST VARIOUS DISTORTIONS. IMAGE-LEVEL F1 AND AUC ARE REPORTED

Distortion	Image-level Detection			
	MVSS-Net		Ours	
	F1	AUC	F1	AUC
w/o Distortion	75.8	**93.7**	**78.0**	88.1
Resize (0.78×)	**76.1**	**88.0**	68.3	72.9
Resize (0.25×)	**42.9**	**80.0**	42.0	68.6
Gaussian Blur (k=3)	**76.3**	**86.2**	64.7	68.4
Gaussian Blur (k=15)	6.9	51.5	**53.6**	**65.0**
Gaussian Noise (σ=3)	0.0	50.0	**76.0**	**87.8**
Gaussian Noise (σ=15)	0.0	50.0	**66.3**	**88.7**
JPEG Compression (q=100)	**63.4**	**88.4**	58.9	77.1
JPEG Compression (q=50)	**45.2**	**74.6**	21.9	63.6
Resize + Gaussian Blur	45.6	**69.1**	**59.6**	66.0
Resize + Gaussian Noise	0.0	50.0	**61.1**	**73.8**
Resize + JPEG Compression	**57.0**	**81.4**	42.4	71.0
Gaussian Blur + Gaussian Noise	0.0	50.0	**22.9**	**62.7**
Gaussian Blur + JPEG Compression	47.9	**68.6**	**48.4**	63.4
Gaussian Noise + JPEG Compression	0.0	50.0	**28.8**	**77.6**
Mixed	**37.1**	57.7	26.1	**65.9**

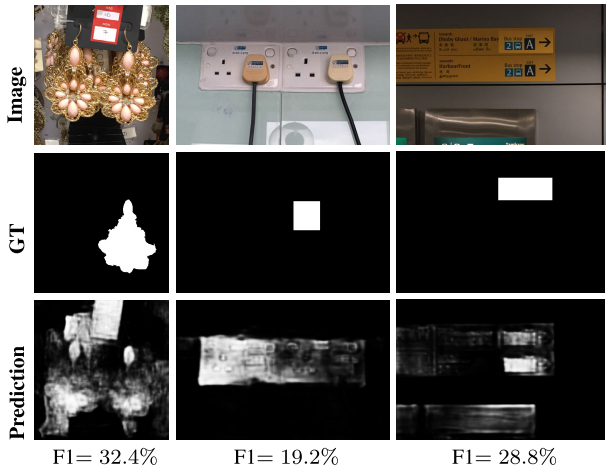

Fig. 12. Failure cases. F1 score for pixel-level localization is presented below each prediction.

Finally, we combine all kinds of the distortions together as "*Mixed*". We apply F1 and AUC to measure the localization performance. Applying distortions to input images will inevitably result in the boundary information being corrupted, thus leading to degraded performance. However, our method shows more robust performance towards the distortions comparing to MVSS-Net, as depicted in Table IX.

We also analyze image-level detection robustness with respect to various distortions on CASIAv1+. As observed in Table X, our performance is inferior to MVSS-Net for resizing and JPEG compression distortion. This could be attributed to the contaminated boundary information caused by these operations, which severely affects our boundary-aware attention module. Nonetheless, our model still achieves reasonable performance, and our robustness is shown to be comparable to, or even superior to that of MVSS-Net for various combinations of attacks.

G. Efficiency Analysis

In terms of computational complexity, our approach achieves 39.44 GFLOPs, which is significantly less than MVSS-Net with 163.57 GFLOPs. Moreover, we assess the computation time using a GeForce RTX 3090 with a 24GB GPU memory. Using the same GPU, our complete model takes 0.014s to process one image, signficantly faster than MVSS-Net, which takes 0.038s.

V. CONCLUSION AND LIMITATIONS

In this work, we aim at the image manipulation localization problem. To this end, we present a novel boundary-guided approach, with an inherent bias towards fully leveraging the boundary information of tampered regions through an attention mechanism and a contrastive learning scheme. In particular, we propose a boundary-aware attention module that predict the boundary of tampered regions to force the network to pay special attention to the important boundary regions. Additionally, we introduce a novel contrastive loss with a boundary-guided sampling strategy to learn more discriminative features. We demonstrate that, as training on CASIAv2, our proposed model outperforms the state-of-the-art methods by a large margin on four different benchmarks. In addition, when pretrained on a synthetic dataset, our model also shows comparable or superior generalizability on real-world benchmarks.

Limitation We show the failure cases on the test set of Coverage in Fig. 12. When the traces of tampered boundaries are carefully erased (the first column of Fig. 12) or similar boundaries are present in both tampered and non-tampered regions (the second column of Fig. 12), our model may have difficulty in discriminating between the boundaries of the tampered regions with those of the non-tampered objects. Moreover, if the tampered regions are local regions of two similar objects (the second and third columns of Fig. 12), our method may potentially fail. For future work, we will develop techniques to distinguish between object boundaries and the boundaries of tampered regions, aiming to further enhance our performance.

In addition, while the main focus of this work is on pixel-level manipulation, we have demonstrated that our model is capable of detecting manipulation at image level. Despite promising performance achieved through simple adaptation of our model to such image-level task, there is definitely some room for further improvement. As a result, an interesting avenue for future research is to specifically tailor our model to the image-level detection problem in order to optimize performance, or explore using our model or part of it as a backbone for joint modeling of the pixel-level and image-level tasks.

REFERENCES

[1] C.-H. Lee, Z. Liu, L. Wu, and P. Luo, "MaskGAN: Towards diverse and interactive facial image manipulation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5549–5558.

[2] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2414–2423, doi: 10.1109/CVPR.2016.265.

[3] X. Liu et al., "Open-Edit: Open-domain image manipulation with open-vocabulary instructions," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 89–106.

[4] N. Kawabata and T. Nakaguchi, "Color laparoscopic image region segmentation after contrast enhancement including SRCNN by image regions," in *Proc. Int. Forum Med. Imag. Asia*, Apr. 2021, pp. 40–45.

[5] X. Chen, C. Dong, J. Ji, J. Cao, and X. Li, "Image manipulation detection by multi-view multi-scale supervision," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 14165–14173.

[6] T. Bianchi and A. Piva, "Image forgery localization via block-grained analysis of JPEG artifacts," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 1003–1017, Jun. 2012.

[7] 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.

[8] G. Li, Q. Wu, D. Tu, and S. Sun, "A sorted neighborhood approach for detecting duplicated regions in image forgeries based on DWT and SVD," in *Proc. IEEE Multimedia Expo. Int. Conf.*, Jul. 2007, pp. 1750–1753.

[9] A. C. Popescu and H. Farid, "Exposing digital forgeries in color filter array interpolated images," *IEEE Trans. Signal Process.*, vol. 53, no. 10, pp. 3948–3959, Oct. 2005, doi: 10.1109/tsp.2005.855406.

[10] 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.

[11] V. V. Kniaz, V. Knyaz, and F. Remondino, "The point where reality meets fantasy: Mixed adversarial generators for image splice detection," in *Proc. NeurIPS*, vol. 32, 2019, pp. 1–11.

[12] Y. Rao and J. Ni, "A deep learning approach to detection of splicing and copy-move forgeries in images," in *Proc. IEEE Int. Workshop Inf. Forensics Security (WIFS)*, Dec. 2016, pp. 1–6.

[13] Y. Wu, W. Abd-Almageed, and P. Natarajan, "Image copy-move forgery detection via an end-to-end deep neural network," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 1907–1915.

[14] Y. Wu, W. Abd-Almageed, and P. Natarajan, "BusterNet: Detecting copy-move image forgery with source/target localization," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 168–184.

[15] X. Zhu, Y. Qian, X. Zhao, B. Sun, and Y. Sun, "A deep learning approach to patch-based image inpainting forensics," *Signal Process., Image Commun.*, vol. 67, pp. 90–99, Sep. 2018.

[16] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *ICML*. PMLR, 2020, pp. 1597–1607.

[17] X. Chen and K. He, "Exploring simple Siamese representation learning," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit.*, Jun. 2021, pp. 15750–15758.

[18] J. Li, P. Zhou, C. Xiong, and S. C. H. Hoi, "Prototypical contrastive learning of unsupervised representations," 2020, *arXiv:2005.04966*.

[19] P. Khosla et al., "Supervised contrastive learning," in *Proc. NIPS*, 2020, pp. 18661–18673.

[20] M. Ye, X. Zhang, P. C. Yuen, and S.-F. Chang, "Unsupervised embedding learning via invariant and spreading instance feature," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6210–6219.

[21] P. Zhou et al., "Generate, segment, and refine: Towards generic manipulation segmentation," in *Proc. AAAI*, 2020, pp. 1–19.

[22] J. Xie, J. Xiang, J. Chen, X. Hou, X. Zhao, and L. Shen, "C2 AM: Contrastive learning of class-agnostic activation map for weakly supervised object localization and semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 989–998.

[23] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9729–9738.

[24] 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.

[25] T.-T. Ng, J. Hsu, and S.-F. Chang. (2009). *Columbia Image Splicing Detection Evaluation Dataset*. DVMM lab. Columbia Univ CalPhotos Digit Libr. [Online]. Available: https://www.ee.columbia.edu/ln/dvmm/downloads/authsplcuncmp/

[26] B. Wen, Y. Zhu, R. Subramanian, T.-T. Ng, X. Shen, and S. Winkler, "COVERAGE—A novel database for copy-move forgery detection," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 161–165.

[27] Nist. (2016). *Nimble 2016 Datasets*. [Online]. Available: https://www.nist.gov/itl/iad/mig/nimble-challenge-2017-evaluation

[28] J. Li, X. Li, B. Yang, and X. Sun, "Segmentation-based image copy-move forgery detection scheme," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 3, pp. 507–518, Mar. 2015.

[29] 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.

[30] 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.

[31] Y. Niu, B. Tondi, Y. Zhao, R. Ni, and M. Barni, "Image splicing detection, localization and attribution via JPEG primary quantization matrix estimation and clustering," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 5397–5412, 2021.

[32] 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.

[33] L. Zhuo, S. Tan, B. Li, and J. Huang, "Self-adversarial training incorporating forgery attention for image forgery localization," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 819–834, 2022.

[34] J. Wang et al., "ObjectFormer for image manipulation detection and localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 2354–2363.

[35] R. Salloum, Y. Ren, and C.-C. J. Kuo, "Image splicing localization using a multi-task fully convolutional network (MFCN)," *J. Vis. Commun. Image Represent.*, vol. 51, pp. 201–209, Feb. 2018.

[36] 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.

[37] 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. 9535–9544.

[38] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, "Learning rich features for image manipulation detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1053–1061.

[39] 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.

[40] R. Han, X. Wang, N. Bai, Q. Wang, Z. Liu, and J. Xue, "FCD-Net: Learning to detect multiple types of homologous deepfake face images," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 2653–2666, 2023.

[41] C. Zhao, C. Wang, G. Hu, H. Chen, C. Liu, and J. Tang, "ISTVT: Interpretable spatial–temporal video transformer for deepfake detection," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 1335–1348, 2023.

[42] Z. Yang, J. Liang, Y. Xu, X.-Y. Zhang, and R. He, "Masked relation learning for deepfake detection," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 1696–1708, 2023.

[43] J. Yang, A. Li, S. Xiao, W. Lu, and X. Gao, "MTD-Net: Learning to detect deepfakes images by multi-scale texture difference," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 4234–4245, 2021.

[44] C. Miao, Z. Tan, Q. Chu, N. Yu, and G. Guo, "Hierarchical frequency-assisted interactive networks for face manipulation detection," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 3008–3021, 2022.

[45] W. Yang et al., "AVoiD-DF: Audio-visual joint learning for detecting deepfake," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 2015–2029, 2023.

[46] F. Wang, H. Liu, D. Guo, and S. Fuchun, "Unsupervised representation learning by invariance propagation," in *Proc. NeurIPS*, vol. 33, 2020, pp. 3510–3520.

[47] J. Robinson, C.-Y. Chuang, S. Sra, and S. Jegelka, "Contrastive learning with hard negative samples," 2020, *arXiv:2010.04592*.

[48] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 2006, pp. 1735–1742.

[49] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3733–3742.

[50] X. Ji, A. Vedaldi, and J. Henriques, "Invariant information clustering for unsupervised image classification and segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9865–9874.

[51] C. Doersch and A. Zisserman, "Multi-task self-supervised visual learning," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2051–2060.

[52] F. F. Niloy, K. Kumar Bhaumik, and S. S. Woo, "CFL-Net: Image forgery localization using contrastive learning," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 4642–4651.
[53] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[54] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 3–19.
[55] A. Vaswani et al., "Attention is all you need," in *Proc. NeurIPS*, vol. 30, 2017, pp. 1–17.
[56] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7794–7803.
[57] K. Norifumi, "3D CG image region of interest estimation and visual attention based on saliency map," in *Proc. Int. Display Workshops*, 2022, p. 709.
[58] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. 18th Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, vol. 9351. Cham, Switzerland: Springer, 2015, pp. 234–241.
[59] A. Kirillov, Y. Wu, K. He, and R. Girshick, "PointRend: Image segmentation as rendering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 9799–9808.
[60] X. Li et al., "Improving semantic segmentation via decoupled body and edge supervision," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 435–452.
[61] X. Li et al., "PointFlow: Flowing semantics through points for aerial image segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 4217–4226.
[62] T. Takikawa, D. Acuna, V. Jampani, and S. Fidler, "Gated-SCNN: Gated shape CNNs for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5229–5238.
[63] J. Wei, S. Wang, and Q. Huang, "FNet: Fusion, feedback and focus for salient object detection," in *Proc. AAAI Conf. Artif. Intell.*, Feb. 2020, vol. 34, no. 7, pp. 12321–12328.
[64] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[65] W. Liu, A. Rabinovich, and A. C. Berg, "ParseNet: Looking wider to see better," 2015, *arXiv:1506.04579*.
[66] X. Hu, Z. Zhang, Z. Jiang, S. Chaudhuri, Z. Yang, and R. Nevatia, "Span: Spatial pyramid attention network for image manipulation localization," in *Proc. ECCV*, 2020, pp. 1–10.
[67] G. Mahfoudi, B. Tajini, F. Retraint, F. Morain-Nicolier, J. L. Dugelay, and P. Marc, "DEFACTO: Image and face manipulation dataset," in *Proc. Eur. Signal Process. Conf.*, 2019, pp. 1–5.
[68] A. Novozámský, B. Mahdian, and S. Saic, "IMD2020: A large-scale annotated dataset tailored for detecting manipulated images," in *Proc. IEEE Winter Appl. Comput. Vis. Workshops (WACVW)*, Mar. 2020, pp. 71–80.
[69] X. Liu, Y. Liu, J. Chen, and X. Liu, "PSCC-Net: Progressive spatio-channel correlation network for image manipulation detection and localization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 11, pp. 7505–7517, Nov. 2022.
[70] T. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.
[71] H. Li and J. Huang, "Localization of deep inpainting using high-pass fully convolutional network," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2019, pp. 8301–8310.
[72] C. Yang, H. Li, F. Lin, B. Jiang, and H. Zhao, "Constrained R-CNN: A general image manipulation detection model," in *Proc. IEEE Int. Conf. Multimedia Expo. (ICME)*, Jul. 2020, pp. 1–6.
[73] M. Kwon, I. Yu, S. Nam, and H. Lee, "CAT-Net: Compression artifact tracing network for detection and localization of image splicing," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 375–384.
[74] C. Dong, X. Chen, R. Hu, J. Cao, and X. Li, "MVSS-Net: Multi-view multi-scale supervised networks for image manipulation detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3539–3553, Mar. 2023.
[75] Y. Zhai, T. Luan, D. Doermann, and J. Yuan, "Towards generic image manipulation detection with weakly-supervised self-consistency learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 22390–22400.
[76] D. Li, J. Zhu, M. Wang, J. Liu, X. Fu, and Z.-J. Zha, "Edge-aware regional message passing controller for image forgery localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 8222–8232.
[77] X. Guo, X. Liu, Z. Ren, S. Grosz, I. Masi, and X. Liu, "Hierarchical fine-grained image forgery detection and localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 3155–3165.
[78] J. Z. Wang, J. Li, and G. Wiederhold, "SIMPLIcity: Semantics-sensitive integrated matching for picture libraries," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 9, pp. 947–963, Jul. 2001.

Wenxi Liu (Member, IEEE) received the Ph.D. degree from the City University of Hong Kong in 2014. He is a Professor with the College of Computer and Data Science, Fuzhou University, China. His research interests include crowd analysis, image segmentation, and image processing.

Hao Zhang is currently pursuing the master's degree with the College of Computer and Data Science, Fuzhou University, China. Her primary research interest is in computer vision.

Xinyang Lin, photograph and biography not available at the time of publication.

Qing Zhang received the M.E. degree from Fuzhou University in 2024. Her research interests lies primarily in image segmentation.

Qi Li received the B.E. degree from Fuzhou University, China, where he is currently pursuing the Ph.D. degree with the College of Computer and Data Science. His research interest lies primarily in computer vision.

Xiaoxiang Liu received the B.E. degree from Fuzhou University, China. He is currently a Postgraduate Student with the College of Computer and Data Science, Fuzhou University. His research interest lies primarily in computer vision.

Ying Cao received the B.Eng. and M.Sc. degrees in software engineering from Northeastern University, China, and the Ph.D. degree in computer science from the City University of Hong Kong. His research interests lie in computer graphics and computer vision.