Article

Patterns

Exposing image splicing traces in scientific publications via uncertainty-guided refinement

Highlights

- URN detects image splicing with high accuracy in scientific publications
- SciSp dataset: large and diverse for spliced scientific images
- Robustness and generalizability of URN testified to through comprehensive experiments
- UGGC and UEMA modules enhance detection performance by fully leveraging uncertainty

Authors

Xun Lin, Wenzhong Tang, Haoran Wang, Yizhong Liu, Yakun Ju, Shuai Wang, Zitong Yu

Correspondence

wangshuai@buaa.edu.cn (S.W.),
yuzitong@gbu.edu.cn (Z.Y.)

In brief

Image manipulation in scientific publications, particularly image splicing, threatens research integrity. Lin et al. introduce the uncertainty-guided refinement network (URN) to detect splicing in scientific images. By leveraging uncertainty information, the URN effectively refines predictions, improving detection accuracy and robustness. The newly developed SciSp dataset, which is large and diverse, supports this advancement. This study has the potential to encourage the integrity of scientific research, contributing to maintaining scholarly credibility and public trust in scientific findings.

Lin et al., 2024, Patterns *5*, 101038
September 13, 2024 © 2024 The Author(s). Published by Elsevier Inc.
https://doi.org/10.1016/j.patter.2024.101038

CellPress

Patterns

CellPress
OPEN ACCESS

Article

Exposing image splicing traces in scientific publications via uncertainty-guided refinement

Xun Lin,[1] Wenzhong Tang,[1] Haoran Wang,[1] Yizhong Liu,[2] Yakun Ju,[3] Shuai Wang,[1,5,*] and Zitong Yu[4,*]
[1]School of Computer Science and Engineering, Beihang University, Beijing 100191, China
[2]School of Cyber Science and Technology, Beihang University, Beijing 100191, China
[3]School of Computer, Ocean University of China, Qingdao 266100, China
[4]School of Information Science and Technology, Great Bay University, Dongguan 523000, China
[5]Lead contact
*Correspondence: wangshuai@buaa.edu.cn (S.W.), yuzitong@gbu.edu.cn (Z.Y.)
https://doi.org/10.1016/j.patter.2024.101038

THE BIGGER PICTURE The integrity of scientific images faces scrutiny due to rising manipulations, leading to retractions. Detecting subtle manipulations like image splicing is difficult when there are no reference images and when there are disruptive factors such as artifacts, abnormal patterns, and noise. Our study introduces the uncertainty-guided refinement network (URN), a robust framework to detect splicing in scientific images. By using uncertainty information, the URN is able to minimize the impact of disruptive factors and improve the accuracy and robustness of its predictions, even after postprocessing. Additionally, we created the SciSp dataset, a comprehensive collection of spliced scientific images, which can be a valuable resource for this field. This work has the potential to discourage fraudulent imagery in the scholarly community and enhance public trust in scientific research.

SUMMARY

Recently, a surge in image manipulations in scientific publications has led to numerous retractions, highlighting the importance of image integrity. Although forensic detectors for image duplication and synthesis have been researched, the detection of image splicing in scientific publications remains largely unexplored. Splicing detection is more challenging than duplication detection due to the lack of reference images and more difficult than synthesis detection because of the presence of smaller tampered-with areas. Moreover, disruptive factors in scientific images, such as artifacts, abnormal patterns, and noise, present misleading features like splicing traces, rendering this task difficult. In addition, the scarcity of high-quality datasets of spliced scientific images has limited advancements. Therefore, we propose the uncertainty-guided refinement network (URN) to mitigate these disruptive factors. We also construct a dataset for image splicing detection (SciSp) with 1,290 spliced images by collecting and manually splicing. Comprehensive experiments demonstrate the URN's superior splicing detection performance.

INTRODUCTION

In scientific publications, erroneous conclusions and irreproducible experimental results from inappropriate image manipulation pose significant threats to the scholarly community.[1] Such manipulations can mislead academic peers, lead to a severe waste of resources, and significantly decrease public confidence in scientific research.[2] Especially in biomedical research, accurate and reproducible data are crucial for advancing medical knowledge and developing treatments for the betterment of humanity.

Scientific images are essential components of scientific publications, widely used to present experimental results.[3] As shown in Figure 1A, malicious researchers can manipulate images to draw experimental conclusions or conceal unfavorable results,[4] posing significant threats to the scholarly community.[1] Given that scientific images usually involve highly specialized and complex experiments, it becomes difficult for readers and reviewers to verify their authenticity through replication promptly.

With advancements in deep learning techniques and the availability of scientific forensic image datasets,[6–8] notable progress has been made in detecting image duplications,[8–10] synthesis,[11,12]

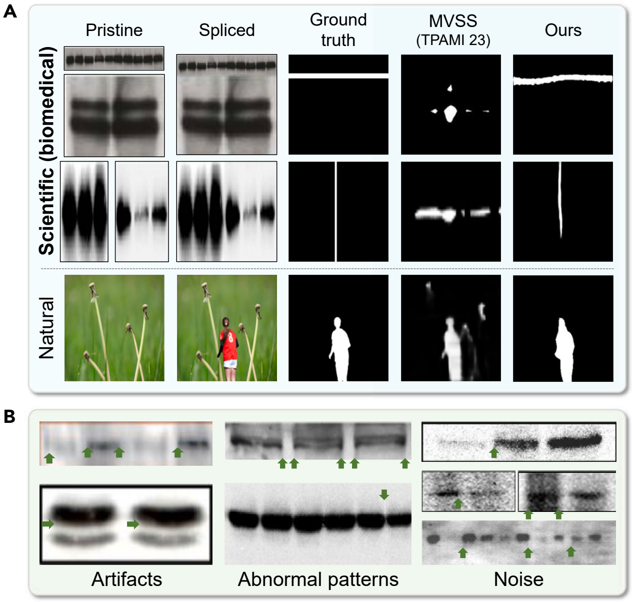

Figure 1. Challenges of detecting splicing traces in scientific images

(A) Difference between spliced scientific (top two rows) and natural images (bottom row). Examples of splicing detection results on the two kinds of images predicted by our method and a famous natural image manipulation detector, i.e., MVSS.[5]

(B) Disruptive factors such as artifacts, abnormal patterns, and noise in scientific images. The green arrows indicate regions interfered with by disruptive factors, whereas the competing manipulation detectors designed for natural images tend to give false alarms.

and copy-move manipulation.[9] However, splicing, one of the most commonly used manipulations for scientific images, is more challenging to detect and has received less attention. Compared to duplication and copy-move detection, splicing detection lacks sufficient references for comparison and retrieval. Furthermore, image synthesis, in contrast to image splicing, tends to generate more detectable forgery cues. This is because image synthesis typically produces global forgeries, affecting the entire image, whereas image splicing results in localized forgeries, affecting specific parts of the image. To the best of our knowledge, no deep-learning-based method has been proposed for scientific image splicing detection.

However, the detection of manipulated (including spliced) pixels in natural images captured by digital cameras has been studied extensively, and several standard datasets have been proposed for this purpose.[13] In the early years, with advances in deep learning techniques, GSR[14] and RRU[15] introduced end-to-end convolutional neural networks (CNNs) for natural image manipulation detection. The effectiveness of self-attention mechanisms in this task has also been discussed in TransForensics[16] and PSCC-Net.[17] The analysis of intrinsic statistics to explore traces of image forgeries has also progressed.[18–20] For more generalized detection, many methods employ noise-sensitive filters[5,21–24] or self-supervised learning[25,26] to suppress semantic information and analyze noise inconsistencies in the images. More recently, Hifi[27] expanded this task and proposed fine-grained image forgery detection with hierarchical labels. This appears to be a feasible way to train the aforementioned methods using scientific datasets. However, existing natural image manipulation detectors, whether proposed for splicing detection[15] or general purposes,[5,14,17,19,22,26,27] cannot achieve satisfactory performance on scientific datasets.[6]

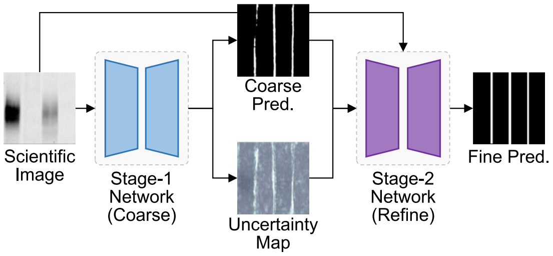

Figure 2. Overall structure of the proposed URN

The structure comprises two stages: a stage 1 network makes coarse predictions of spliced regions and estimates the pixel-level uncertainty about coarse predictions and a stage 2 uncertainty-guided refinement network can refine coarse predictions under the guidance of uncertainty information.

We attribute this failure to two primary reasons: (1) the prevalence of more disruptive factors in scientific images and (2) the limited number of spliced images for training. Disruptive factors mislead detection methods, rendering false alarms and incomplete predictions. These factors include the following (see also Figure 1B).

(1) Artifacts. Scientific images undergo multiple types of non-malicious degradation by authors and publishers. Common degradations, e.g., JPEG compression,[28] can decrease tampering traces and introduce more artifacts.[29]
(2) Abnormal patterns. During experimental processes, operational mistakes (e.g., improper reagent ratios and gel rupture) can result in abnormal patterns, which tend to be confused with splicing traces.
(3) Noise. Malfunctions and degradations of imaging devices or even simple operator errors can introduce a large amount of noise.

Furthermore, malicious researchers may use advanced image editing tools[30] such as Photoshop[31] or advanced AI-based generative models, e.g., generative adversarial networks[32] and diffusion models,[33] to postprocess the spliced images. These postprocessing approaches can reduce the visible clues of splicing,[34] rendering distinguishing between natural boundaries and splicing traces challenging.

To address these issues, we design a two-stage uncertainty-guided refinement network (URN) that can resist disruptive factors and various postprocessing approaches (see Figure 2). In stage 1, our model recognizes the uncertain predicted regions affected by disruptive factors using Monte Carlo dropout (MCD).[35] In stage 2, the URN integrates the uncertainty to perform refinement. To exploit the uncertainty information fully, we propose uncertainty-guided graph convolution (UGGC) modules. UGGC limits unreliable information flow from uncertain regions but guides them to receive information from nearby confident regions. This ensures that uncertain nodes can be gradually refined with less interference. Additionally, we propose

Table 1. Details of datasets for scientific image splicing detection

Name	No. of spliced images	Type	No. of splicing approaches	Acquisition methods	Sources
Biofors[6]	181	blot/gel	3	found in publications	from *PLOS One*
RSIID[7]	880	microscopy	1	automatically spliced	from 3 public datasets
SciSp-C	290	blot/gel	3	found in publications	from 39 journals
SciSp-H	1,000	blot/gel	5	manually spliced	from *PLOS One*

uncertainty-enhanced manipulation attention (UEMA) modules that can focus on ambiguously predicted regions with high uncertainty and help distinguish between spliced and pristine areas. By combining UGGC and UEMA, the performance and robustness of our URN can be further improved.

In addition to effective methodologies, high-quality datasets are essential for this task. Existing works,[7] BINDER,[8] and SYB[36] automated the forgery of scientific images to generate large-scale forensics datasets, which are designed for the detection of manipulation, duplication, and synthesis, respectively. To prevent the adverse effects introduced by the domain gap between automatically generated images and real-world ones, Biofors[6] extracted 47,805 scientific images from papers published in *PLOS ONE* 2013, including 1,741 manipulated images with pixel-level labels according to raw annotations provided by academic experts.[2] Table 1 shows that the number of spliced images remains insufficient, particularly in Biofors, which contains only 181 blot/gel images. Methods trained on these datasets tend to have limited generalizability and cannot be directly applied to real-world applications. Sufficient datasets are available for detecting scientific image duplications and copy-move manipulations. These datasets have led to significant progress being made in these two tasks.[9,10] However, the lack of high-quality datasets for scientific image splicing detection has hindered progress in this field. In particular, the lack of diversity in splicing approaches and image sources hampers the training of models with promising performance.

Therefore, we construct a dataset for scientific image splicing detection (SciSp), which is a diverse and challenging dataset that surpasses others in terms of size, source diversity, and splicing complexity. SciSp comprises two subsets: one collected from public comments on the authoritative post-publication peer review platform PubPeer (SciSp-C) and the other comprising images manually manipulated with Photoshop (SciSp-H). Our contributions can be summarized as follows.

(1) We propose an end-to-end deep-learning-based framework for scientific image splicing detection. Our URN can resist disruptive factors in scientific images by using uncertainties to refine coarse predictions.
(2) We propose two modules, i.e., UGGC and UEMA, to help the URN integrate uncertainty information. UGGC can extract robust features against disruptive factors during encoding, whereas UEMA focuses on the refinement of uncertainly predicted areas in the decoding phase.
(3) We introduce a dataset for scientific image splicing (SciSp) detection. Compared with existing datasets, SciSp has more spliced images from diverse sources and various splicing approaches.
(4) We conduct comprehensive experiments on four benchmarks, demonstrating the efficacy of the proposed method in both pixel-level and image-level detection.

RESULTS

We use the F1-score (F1) and Matthews correlation coefficient (MCC)[37] for pixel-level evaluation. For image-level assessment, we adopt the area under the receiver operating characteristic curve (AUC) and accuracy (Acc).

Scientific image splicing detection performance

To the best of our knowledge, no deep-learning-based method has been proposed for this task. Therefore, we adopt ten deep-learning-based image manipulation or splicing detection methods designed for natural images for comparisons. Notably, all the selected image manipulation methods can be used for splicing detection. These methods fall into two categories: (1) noise-sensitive methods, including ManTra,[22] MVSS,[5] and TruFor,[26] and (2) noise-insensitive methods, i.e., RRU-Net,[15] HDU-Net,[38] GSR-Net,[14] IF-OSN,[29] PSCC-Net,[17] CAT-Net 2,[19] and Hifi.[27] All these methods have publicly available source codes and were retrained on selected datasets. We automatically chose their hyperparameters as described in the corresponding reference papers or optimally assigned the better ones. We also compare with a handcrafted-feature-based method, namely FBIGEL,[39] which is designed for blot/gel image splicing detection. For methods that require additional detection capabilities from external datasets, such as TruFor, which learns camera fingerprints, and IF-OSN, which learns the degradation distribution of social networks, we fixed the official pretrained weights for the specific parts according to their instructions. The hyperparameters of each competing method are listed in Table S4. We also present the comparison results on natural images (NIST 2016[40]) in Table S1.

In Table 2, we present both the pixel- and image-level detection results. Our URN outperforms all competing methods in terms of average metric values across all datasets, achieving the highest scores in pixel-level F1 and MCC for each dataset. Additionally, approximately half of the image-level metric values of the URN ranked second. The proposed method has fewer false alarms and is more capable of accurately localizing subtle splicing traces (see Figure 3). This indicates the effectiveness of our strategy of using uncertainty information to refine predictions. Besides, images in RSIID have fewer disruptive factors,

Table 2. Pixel-level (i.e., F1 and MCC) and image-level (i.e., AUC and Acc) performance (%) of scientific image splicing detection

Datasets →	Biofors				RSIID				SciSp-C				SciSp-H				Average			
Methods ↓	F1	MCC	AUC	Acc	F1	MCC	AUC	Acc	F1	MCC	AUC	Acc	F1	MCC	AUC	Acc	F1	MCC	AUC	Acc
FBIGEL[39]	4.1	0.0	50.0	50.0	5.7	0.0	50.0	50.0	0.3	0.0	50.0	50.0	7.91	0.0	50.0	31.2	4.5	0.0	50.0	45.3
RRU-Net[15]	14.9	14.9	79.9	75.5	80.6	80.4	99.1	94.0	31.9	32.7	83.0	73.3	32.4	32.8	76.1	68.0	40.0	40.2	83.4	77.7
HDU-Net[38]	20.6	22.0	85.2	75.0	80.1	80.6	99.2	96.2	33.6	34.3	84.2	72.7	36.1	36.5	74.6	64.8	42.6	43.0	83.3	77.2
ManTra*[22]	12.4	8.0	55.3	50.0	48.8	49.9	98.4	84.1	20.7	21.3	73.8	63.4	0.4	−0.1	54.8	59.8	20.6	20.9	69.2	64.3
ManTra[22]	9.0	8.0	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
GSR-Net[14]	3.1	2.4	70.3	65.7	79.1	79.0	98.8	93.3	21.7	22.2	73.4	65.8	23.6	23.4	71.1	66.8	31.9	31.9	77.2	72.9
MVSS[5]	11.4	10.9	77.8	60.0	77.3	77.2	99.3	95.1	15.8	16.2	79.9	64.2	29.0	29.0	76.7	66.8	33.4	33.5	79.0	71.5
IF-OSN[29]	9.6	8.7	72.8	66.7	80.1	80.0	98.9	95.1	14.3	13.5	76.7	69.8	12.5	12.0	65.3	58.0	29.1	28.8	76.9	72.4
PSCC-Net[17]	12.9	7.0	71.2	63.0	70.1	70.3	99.5	95.4	6.5	1.5	79.4	68.0	18.3	14.7	61.9	31.2	27.0	24.9	76.0	64.4
CAT-Net 2[19]	13.8	12.4	59.8	53.7	77.4	77.1	97.9	91.2	22.9	23.0	76.8	69.8	30.4	30.3	74.6	64.0	36.1	36.1	75.8	69.7
TruFor[26]	8.3	8.9	71.4	66.7	77.3	77.5	99.3	94.7	17.5	17.6	75.8	64.0	19.7	19.6	74.0	65.3	30.7	30.8	79.0	72.7
Hifi[27]	21.7	20.1	66.7	61.1	77.4	77.2	98.7	84.5	35.8	37.7[b]	77.2	72.7	26.2	26.0	66.8	69.0	40.3	40.7	75.9	71.8
URN	30.3	30.4	87.8	75.0	81.4	81.4	99.3	94.9	38.5	39.3	85.9	75.0	38.9	38.9	78.0	68.7	47.3	47.5	84.6	78.4

"ManTra" represents the results reported in Biofors,[6] whereas "ManTra*" denotes the results we reproduce.

such as noise and artifacts, which are less likely to mislead the detection method into incorrect predictions. The superior performance of the proposed method on SciSp-H further demonstrates that the effective use of uncertainty can mitigate the effects of disruptive factors, thereby enhancing robustness. It also can be seen from Table 2 that FBIGEL performs poorly across all datasets, despite being designed for gel image splicing detection. FBIGEL's performance is significantly worse than deep-learning-based methods because it does not learn to extract generalized splicing features from a sufficient amount of images.

Robustness against postprocessing

In this section, we describe the experiments conducted to verify the robustness of our method in handling different forms of postprocessing, that is, degradation and inpainting. For fair verification, we utilized only postprocessing approaches for the testing set.

Capability to resist degradation

In real-world scenarios, scientific images can be degraded using multiple approaches. We simulated these conditions using various degrees of commonly used degradations. For each degradation approach, we applied various parameter values, including the kernel size of Gaussian blur (from 3 to 13), the quality of JPEG compression (from 15% to 90%), and the standard deviation of Gaussian noise (from 5 to 30), to perform a comprehensive verification. Additionally, we evaluated the performance of images recaptured using the Windows Snipping Tool on a Windows 10 computer with a screen resolution of 1,920 × 1,080. As observed in Figure 4, the proposed method maintains its best performance under various degradation attacks. Even under the most severe degradation, our performance exceeds that of the other methods with no degradation.

Capability to resist inpainting

Inpainting is widely used for obtaining hidden visual clues during image manipulation.[41] To validate the resistance of the models to inpainting attacks, we employed two state-of-the-art generative algorithms, i.e., LaMa[32] and Stable Diffusion v.2[33] and two commonly used traditional algorithms, Navier-Stokes[42] and Telea,[43] to conceal the spliced regions (see Figure 5). To the best of our knowledge, this type of experiment has not been previously performed in the field of image manipulation detection. As shown in Table 3, our proposed approach achieves the best overall performance when using various inpainting methods.

Generalizability against domain shifts

However, in real-world applications, these splicing detection methods are challenged by domain shifts and unseen splicing. To further validate the generalizability, we conducted cross-dataset testing. As shown in Table 4, our URN achieves the best performance in terms of the average metric values for all protocols. The URN also reaches the highest scores on half of the metrics and ranks second on the rest. This indicates that our method can resist domain shifts and unseen attacks using reliable information from the confident regions.

DISCUSSION

Figure 4 and Tables 2, 3, and 4 demonstrate the outstanding detection performance, robustness, and generalizability of the

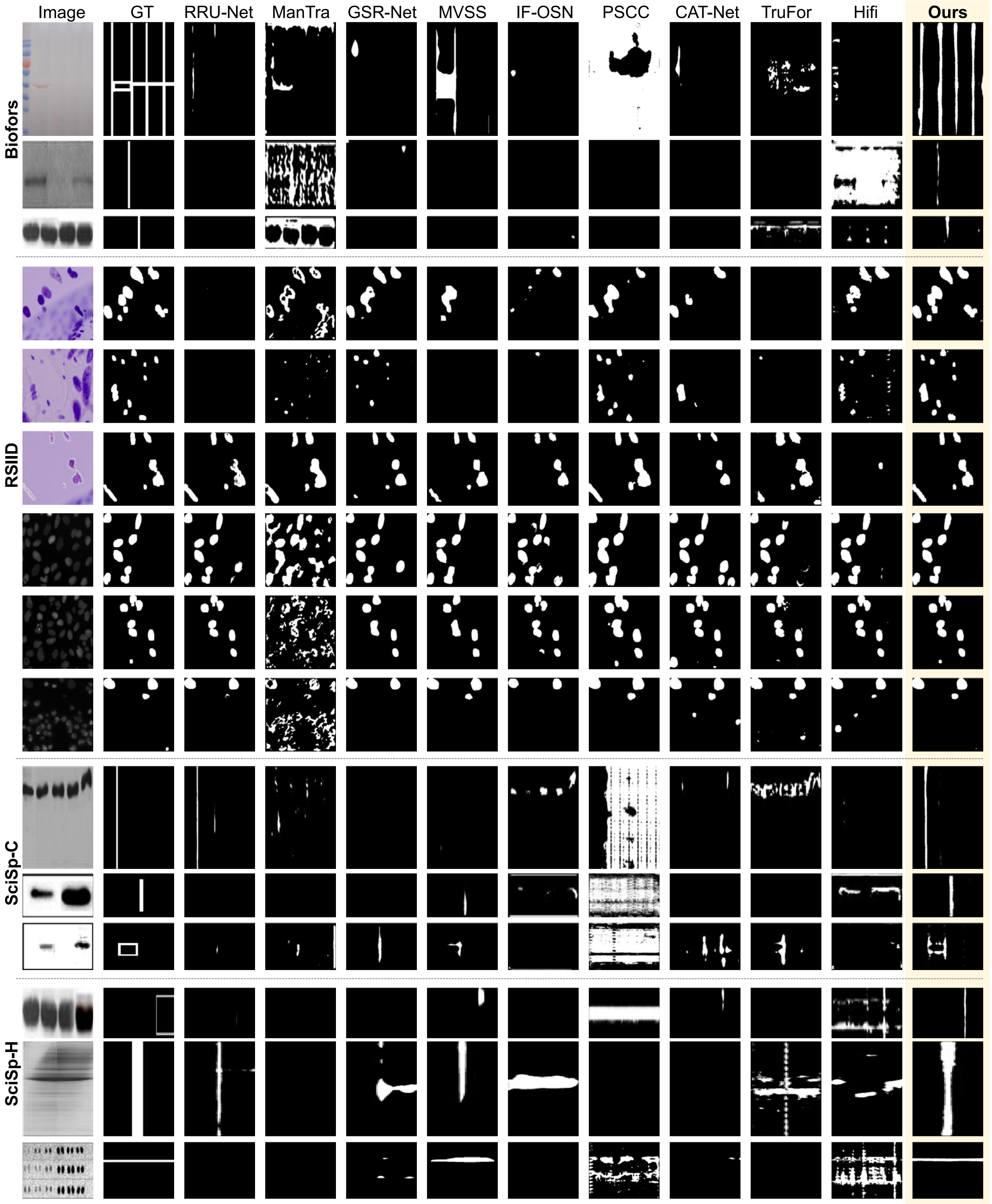

Figure 3. Qualitative results across Biofors, RSIID, SciSp-C, and SciSp-H
From left to right, each column represents spliced images, ground truths (GTs), and the predictions of competing methods, respectively.

URN. This can be attributed to the effective utilization of uncertainty information. Even if the confidence of the predictions decreases in some areas, our stage 2 network integrated with the proposed UEMA and UGGC can refine them to yield inspiring results. In the following subsections, we describe the ablation study, the URN's limitations, and the social impacts of this study.

Ablation study on URN

To comprehensively validate the individual effectiveness of the components of the URN, we not only compared the performance of our model before and after the inclusion of critical modules but also conducted comparisons between multiple variants of key submodules. In Table 5, we present the specific configurations and performance of different variants by replacing or removing

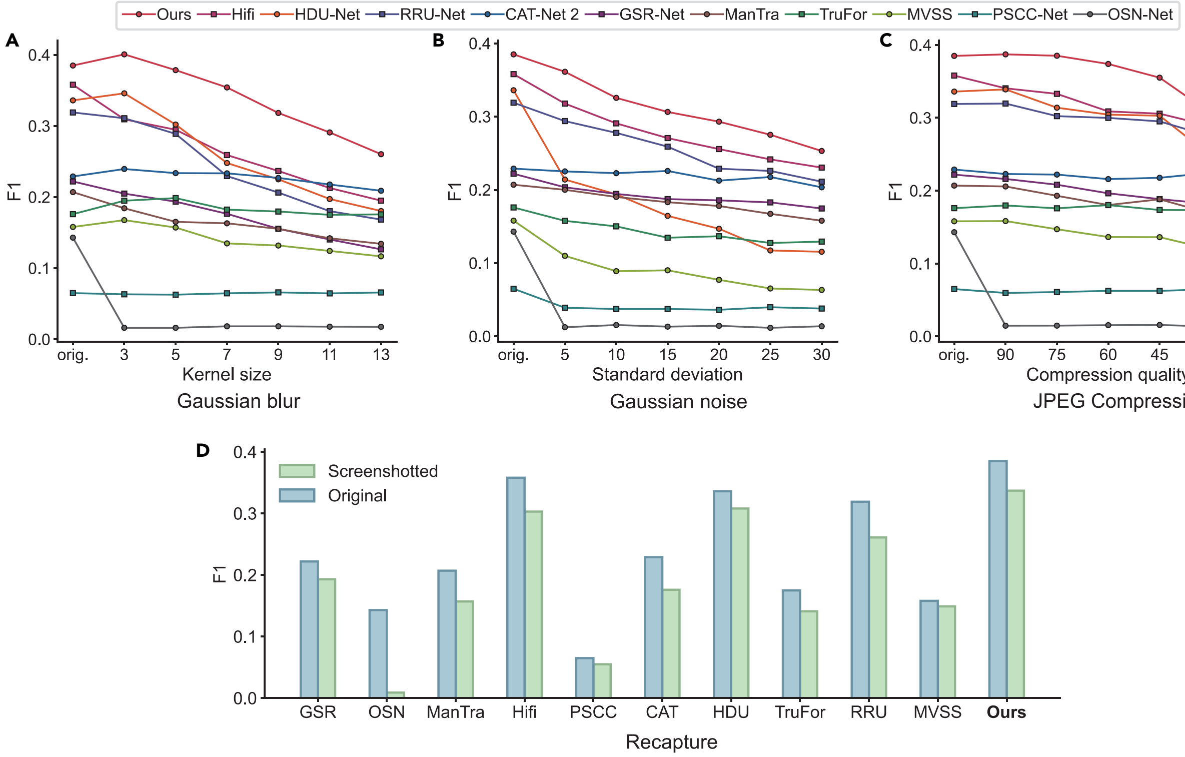

Figure 4. Robustness against image degradations

(A–D) Robustness analysis results on SciSp-C against four different degradations, i.e., (A) Gaussian blur, (B) Gaussian noise, (C) JPEG compression, and (D) recapture by Windows Snipping Tool.

key components. To further confirm the regions each case tends to focus on, we show feature visualization results of all cases using Grad-CAM[44] in Figure 6. The plugin capability of UGGC and UEMA is presented in Table S2.

Effectiveness of the two-stage framework

From row "UM" in Figure 6, it can be seen that the uncertainty maps we estimated in the stage 1 network effectively identify regions prone to erroneous prediction due to disruptive factors. As shown in Table 5, the metric values of case 1 are lower than most of the other cases. Additionally, from the 4th row in Figure 6, we can see that case 1 loses focus on some spliced regions and incorrectly attends to pristine areas. This indicates that appropriately leveraging uncertainty information can improve performance.

Effectiveness of the UGGC

As shown in Table 5, we observe that adding UGGC modules (compare case 2 with "full") brings 6.5% and 4.8% increases in F1 and MCC, respectively. Moreover, it helps to reduce the erroneous attention paid to non-spliced regions with high uncertainty (see Figure 6). This demonstrates that UGGC can assist the URN in more robust learning features. In addition, in cases 3–8, we investigate the impact of different edge connection strategies on the construction of $\mathcal{G}^k$ in UGGC, including the scope of the edge connection and whether the edges are directed and weighted. As shown in Table 5, regardless of the type of edge connection strategy used, all resulted in a decrease in performance compared to the proposed plan. This demonstrates that restricting the information flow from confident nodes to local uncertain nodes helps weakly predicted regions capture local inconsistencies from nearby areas, thereby enhancing the robustness.

Effectiveness of the UEMA

As shown in Table 5 and Figure 6, we explored the effectiveness of UEMA by constructing cases 9–12. Regardless of whether UEMA is removed or replaced with self-attention, cross-attention, or depth-wise convolution,[46] better performance cannot be achieved. Notably, the metric values of case 11 decreased, indicating that a semantic gap exists between the estimated uncertainty maps and the features in the deep layers. A direct calculation of their correlations would mislead the decoding process. This demonstrates that our strategy of enhancing feature maps with uncertainty maps can help UEMA focus on uncertain regions and distinguish between spliced and pristine areas. To further explore the effectiveness of UEMA in different positions

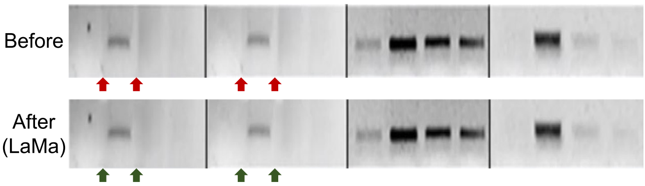

Figure 5. Example of using the advanced AI-based inpainting scheme (LaMa\cite{xxx})[32] to decrease splicing traces in the first two images

The arrows indicate the splicing traces. The source image is from Biofors.

Table 3. Robustness comparisons against four inpainting approaches (i.e., LaMa, Stable Diffusion v.2, Navier-Stokes, and Telea) on SciSp-C

Inpainting →	LaMa[32]				Stable Diffusion v.2[33]				Navier-Stokes[42]				Telea[43]				Average			
Methods ↓	F1	MCC	AUC	Acc	F1	MCC	AUC	Acc	F1	MCC	AUC	Acc	F1	MCC	AUC	Acc	F1	MCC	AUC	Acc
RRU-Net[15]	15.6	15.8	73.9	66.9	31.9	32.7	83.0	73.3	25.2	25.6	79.9	70.4	20.4	21.0	77.1	68.0	23.3	23.8	78.5	69.6
HDU-Net[38]	16.1	16.7	73.0	66.9	33.7	34.9	86.0	73.8	31.8	32.7	85.0	74.4	23.7	24.5	79.1	70.9	26.3	27.2	80.8	71.5
ManTra[15]	8.2	8.2	68.2	62.8	20.7	21.3	73.8	63.4	15.2	15.4	70.7	61.6	11.5	11.5	69.4	62.2	13.9	14.1	70.5	62.5
GSR-Net[14]	12.7	13.2	68.0	66.3	21.7	22.2	73.4	68.6	15.1	15.7	69.5	68.0	13.6	14.4	69.0	67.4	15.7	16.4	70.0	67.6
MVSS[5]	12.1	12.2	75.0	62.8	15.8	16.2	79.9	69.8	13.4	14.4	80.1	68.6	12.4	13.0	78.4	66.9	13.4	14.0	78.3	67.0
IF-OSN[29]	1.0	1.3	68.2	62.8	1.5	1.9	68.2	62.8	1.0	1.2	68.0	62.8	1.0	1.2	67.8	62.8	1.1	1.4	68.1	62.8
PSCC-Net[17]	5.8	2.8	58.2	50.0	5.9	3.1	58.6	50.0	6.1	3.4	58.6	50.0	5.9	3.3	58.2	50.0	5.9	3.1	58.4	50.0
CAT-Net 2[19]	20.9	20.8	75.2	69.2	22.9	23.0	76.8	69.8	21.3	21.4	75.6	70.9	20.4	20.4	75.7	70.9	21.4	21.4	75.8	70.2
TruFor[26]	14.6	14.8	70.2	67.4	17.5	17.6	71.7	68.0	13.1	13.3	69.8	68.0	13.1	13.1	69.1	67.4	14.6	14.7	70.2	67.7
Hifi[27]	23.5	24.1	73.0	50.0	35.8	37.7	73.8	50.0	29.8	32.0	73.8	50.0	28.1	29.9	73.4	50.0	29.3	30.9	73.5	50.0
URN	28.6	28.9	82.1	73.8	40.1	40.6	85.7	75.0	36.0	36.7	84.8	74.4	32.5	32.9	84.2	72.1	34.3	34.8	84.2	73.8

of our URN, we compare cases 13, 14, and full in Table 5 and Figure 6. We tested two variants: case 13 with UEMA in both the decoding and encoding stages and case 14 with UEMA only in the encoding stage. We observed that the performance of case 13 was inferior to that of case full, suggesting that the unrestricted global information transition in UEMA interfered with the robust features refined by UGGC during encoding. Moreover, the performance in case 14 was worse than that in case 13, implying that additional attention to uncertain regions was effective during the decoding stage.

Social impacts

The proposed method and dataset yield both beneficial and adverse societal impacts; we believe the former outweighs the latter. The primary benefit lies in the ability of our process to expose splicing traces in scientific images, prevent misleading academic peers, and maintain scholarly integrity. However, the present Acc of our method remains inadequate, necessitating expert review for the validation of the results in real-world applications. Additionally, false alarms generated by our method may be exploited by malicious individuals to unjustly accuse others of publication misconduct. Meanwhile, our method may compel researchers to meticulously manipulate images using more advanced techniques.

Limitations of the study

The efficiency analysis results in Table S3 show that our URN, due to its two-stage network structure, results in lower detection efficiency and a higher parameter count compared to some of the competing methods. Moreover, as shown in Table 4, the cross-dataset results are not as inspiring as the intra-dataset results (see Table 2), which motivates us to explore techniques to improve domain generalizability. In future research, we aim to further improve the domain generalization ability of our method, enabling it to be easily adapted to other types of scientific images or images from different fields (e.g., material science, chemistry, and mechanical engineering). We also observed a significant imbalance between the number of pristine and spliced images. Effectively leveraging a large number of pristine images is also an exciting direction for further investigation.

Moreover, the experimental results in Table S6 indicate that our URN achieved competitive performance in copy-move and removal detection, although there is still scope for improvement. Given the various forms of academic misconduct, such as image splicing, copy-move, object removal, image duplication, AI-generated images, AI-generated text, and text duplication, these methods pose significant threats to the integrity of scientific research. With the recent rapid development of large language and vision language models, an all-in-one scientific publication detection model that enables the comprehensive detection of various forms of academic misconduct may be feasible.

EXPERIMENTAL PROCEDURES

Resource availability

Lead contact

Further information and requests for resources should be directed to and will be fulfilled by the lead contact, Prof. Shuai Wang (wangshuai@buaa.edu.cn).

Materials availability

This work did not provide new materials.

Table 4. Pixel-level (i.e., F1 and MCC) and image-level (i.e., AUC and Acc) cross-testing results (%)

Protocol →	B + C → H				C + H → B				H + B → C				Average			
Method ↓	F1	MCC	AUC	Acc	F1	MCC	AUC	Acc	F1	MCC	AUC	Acc	F1	MCC	AUC	Acc
RRU-Net[15]	13.1	12.6	56.2	53.8	22.1	23.4	70.8	69.4	21.5	22.6	76.0	66.9	18.9	19.6	67.7	63.4
HDU-Net[38]	10.6	10.5	57.5	52.2	19.7	20.9	81.3	73.2	22.6	23.8	78.2	74.4	17.6	18.4	72.3	66.6
ManTra[22]	9.0	2.5	49.3	31.2	0.2	0.1	50.1	52.8	10.2	8.2	47.6	50.0	6.5	3.6	49.0	44.7
GSR-Net[14]	3.7	3.4	52.8	51.8	15.2	15.4	67.7	63.9	1.8	1.5	60.9	56.4	6.9	6.8	60.4	57.4
MVSS[5]	11.2	11.0	60.5	60.7	20.8	21.3	75.3	69.4	13.7	14.3	79.8	71.5	15.3	15.5	71.9	67.2
IF-OSN[29]	9.2	8.5	51.9	47.0	8.1	7.8	52.5	60.2	4.9	5.2	63.9	59.9	7.4	7.1	56.1	55.7
PSCC-Net[17]	11.8	8.4	54.0	31.2	12.5	5.0	60.4	50.0	28.0	31.4	58.8	50.0	17.4	14.9	57.7	43.7
CAT-Net 2[19]	10.1	8.3	55.1	43.2	11.7	11.8	64.0	59.3	12.8	12.9	65.3	61.6	11.5	11.0	61.5	54.7
TruFor[26]	6.2	−0.4	51.0	33.2	11.0	7.9	47.6	49.1	3.7	1.7	51.9	50.6	7.0	3.1	50.2	44.3
Hifi[27]	17.1	13.2	48.4	31.2	27.7	26.3	61.1	50.0	18.7	22.3	59.2	50.0	21.2	20.6	56.2	43.7
URN[b]	15.1	14.6	59.4	55.8	25.8	26.8	82.9	77.8	23.4	24.0	80.9	75.0	21.4	21.8	74.4	69.5

"X + Y → Z" denotes that the union of X and Y is used for training and Z is adopted as the test set. B, Biofors; C, SciSp-C; H, SciSp-H.

Data and code availability

The source code of our URN is publicly available on GitHub (https://github.com/lxbuaa/URN) and has been archived at Zenodo.[47] The proposed datasets SciSp-C and SciSp-H can be downloaded from Zenodo (https://zenodo.org/records/11066372) and have also been archived at there.[48] Other datasets, i.e., Biofors and RSIID, can be found at their official GitHub repositories (https://github.com/vimal-isi-edu/BioFors and https://github.com/phillipecardenuto/rsiil).

Please note that the images in SciSp-C are sourced from various journals, some of which have restrictive copyright policies. As a result, we are unable to publicly release the original images of SciSp-C. Instead, we provide the DOIs of the publications for these images, links to the relevant PubPeer comments, and detailed image identifiers (e.g., Figure 2A) to ensure reproducibility as much as possible.

Datasets

High-quality datasets are essential for advancing the detection of scientific splicing. Most scientific images have not been proven to be improperly manipulated, even if they are forged, rendering data collection difficult. Here, we constructed two subdatasets: we (1) collected images published in various journals according to public comments from PubPeer and (2) manually spliced images using multiple splicing approaches and performed postprocessing to reduce the visible splicing traces.

SciSp-C: Collected images

We collected 110 public comments posted on PubPeer; 290 images showed signs of splicing. Among these images, 42 were from retracted papers, 102 were from corrected papers, and 27 were from publicly acknowledged splicing. The remaining images were deemed spliced because the authors could not provide solid evidence to refute concerns, such as being unable to

Table 5. Ablation results (%) on SciSp-C

		Stage 2 refinement modules						
		Encoding stage				Decoding stage	Metric values	
Case	Refinement	Module	Local	Directed	Weighted	Module	F1	MCC
1	×	–	–	–	–	–	30.4	31.9
2	✓	–	–	–	–	UEMA	32.0	34.5
3	✓	UGGC (kNN[45])	–	✓		UEMA	30.0	31.0
4	✓	UGGC	✓	×	×	UEMA	21.6	22.6
5	✓	UGGC	✓	✓	×	UEMA	33.8	35.3
6	✓	self-attention	×	×	×	UEMA	32.5	34.9
7	✓	UGGC	×	✓	×	UEMA	33.2	34.3
8	✓	UGGC	×	✓	✓	UEMA	35.8	37.0
9	✓	UGGC	✓	✓	✓	–	32.5	33.6
10	✓	UGGC	✓	✓	✓	Self-Attention	35.0	35.5
11	✓	UGGC	✓	✓	✓	Cross-Attention	29.0	30.4
12	✓	UGGC	✓	✓	✓	Depth-wise Convolution	34.6	35.6
13	✓	UGGC+UEMA	✓	✓	✓	UEMA	36.9	37.3
14	✓	UGGC+UEMA	✓	✓	✓	–	34.6	35.5
Full	✓	UGGC	✓	✓	✓	UEMA	38.5	39.3

"Local," "directed," and "weighted" indicate whether the edges in graph $\mathcal{G}$ constructed in UGGC are local, directed, or weighted, respectively. "Refinement" denotes whether the stage 2 refinement network is adopted.

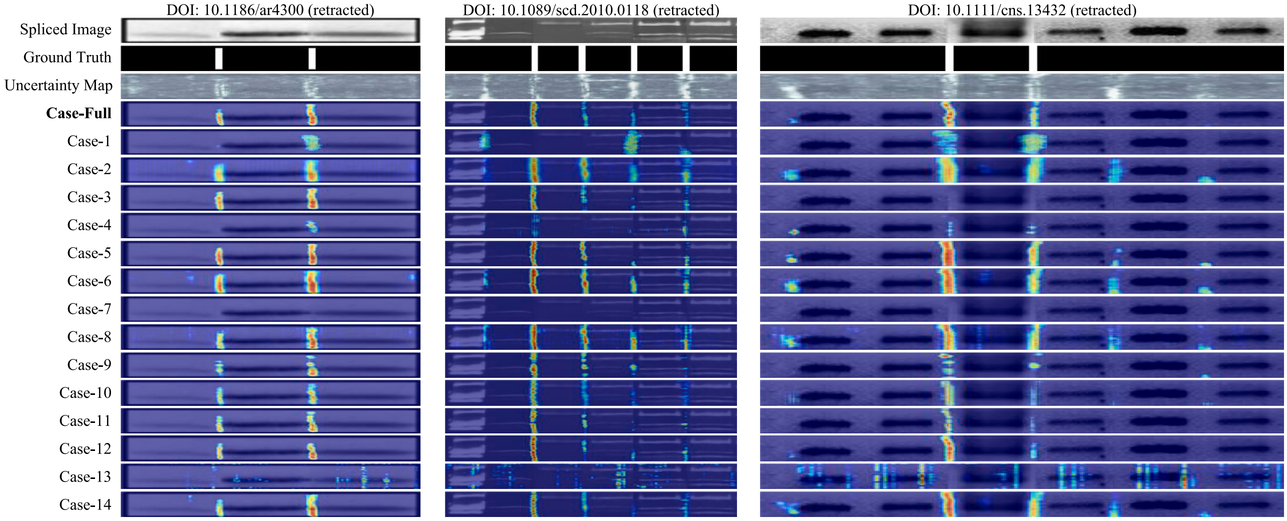

Figure 6. Feature visualization of ablation study

From top to bottom, each row represents the spliced images, GTs, uncertainty maps, and feature heatmaps of the corresponding case.

present the original images. All collected images had raw annotations provided by academic peers, suggesting that the images were highly likely to be spliced.

An example of the annotation process is presented in Figure 7A, where raw annotations were collected from PubPeer. First, we found raw annotations related to the spliced scientific images from public comments provided by experts. To prevent image degradation caused by screenshots, we downloaded the corresponding papers (PDF version) and used Kingsoft WPS (https://www.wps.com/) to extract spliced images. Finally, based on the raw annotations, we use Labelme (https://github.com/wkentaro/labelme) to annotate the pixel-level binary masks of the spliced regions. Given that neither we nor experts know which parts of an image originate from other images, we are unable to annotate all the externally spliced areas, as in natural image datasets.[40] Following the annotation by Biofors, we marked only the junctions of the spliced regions. In contrast, images from RSIID are all automatically generated, with complete external regions annotated in their ground truths. This results in a different form of the ground-truth binary mask compared with Biofors and SciSp. We annotated the splicing traces and nearby regions affected by postprocessing, resulting in thicker bands in the ground-truth masks. In addition, resizing images can cause the traces to become thicker or thinner.

As shown in Figures 7B and 7C, SciSp-C has diverse sources of journals and publication years. It also includes journals from all tiers of influence. Different journals, or the same journal across years, are likely to use different degradation methods. Therefore, expanding the diversity of sources is essential for improving the generalizability of detection models. Most images in SciSp-C were retracted by the publisher or corrected by the authors. For the remaining images, the reviewers provided substantial evidence of traces of splicing.

SciSp-H: Handcrafted images

Given that it is challenging to build a large dataset based on a limited number of publicly questioned images, RSIID automatically splices 880 images according to its preset rules. However, the spliced regions of images in RSIID are easily detectable because the images lack refined postprocessing. Consequently, we manually constructed SciSp-H with 1,000 spliced images to make the dataset more challenging.

To diversify the splicing approaches, we designed five different approaches to generate images, i.e., (1) vertical splicing: splice two images in the vertical direction; (2) horizontal splicing: splice two images in the horizontal direction; (3) free splicing: select a part from one image and then splice it onto another image; (4) vertical removal and splicing: remove a vertical part of an image and then horizontally splice the remaining two parts together; and (5) horizontal removal and splicing: remove a horizontal part of an image and then vertically splice the remaining two parts together. A detailed illustration of the five splicing approaches is presented in Figure 8. For each

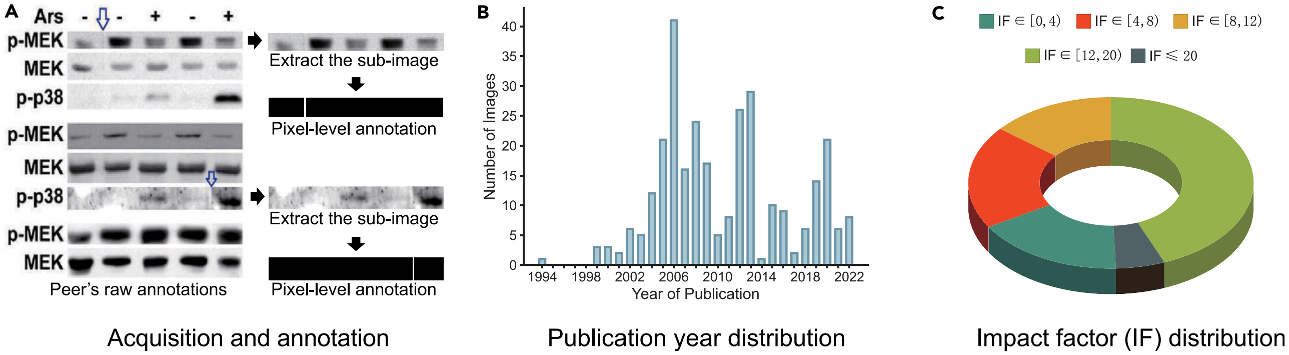

Figure 7. Details of SciSp-C

(A) Left: the academic peer uses blue arrows to indicate splicing traces. Right: we make pixel-level annotations according to the peer's raw annotations.

(B) Distribution of publication years for papers containing SciSp-C images.

(C) Distribution of impact factors for journals containing SciSp-C images.

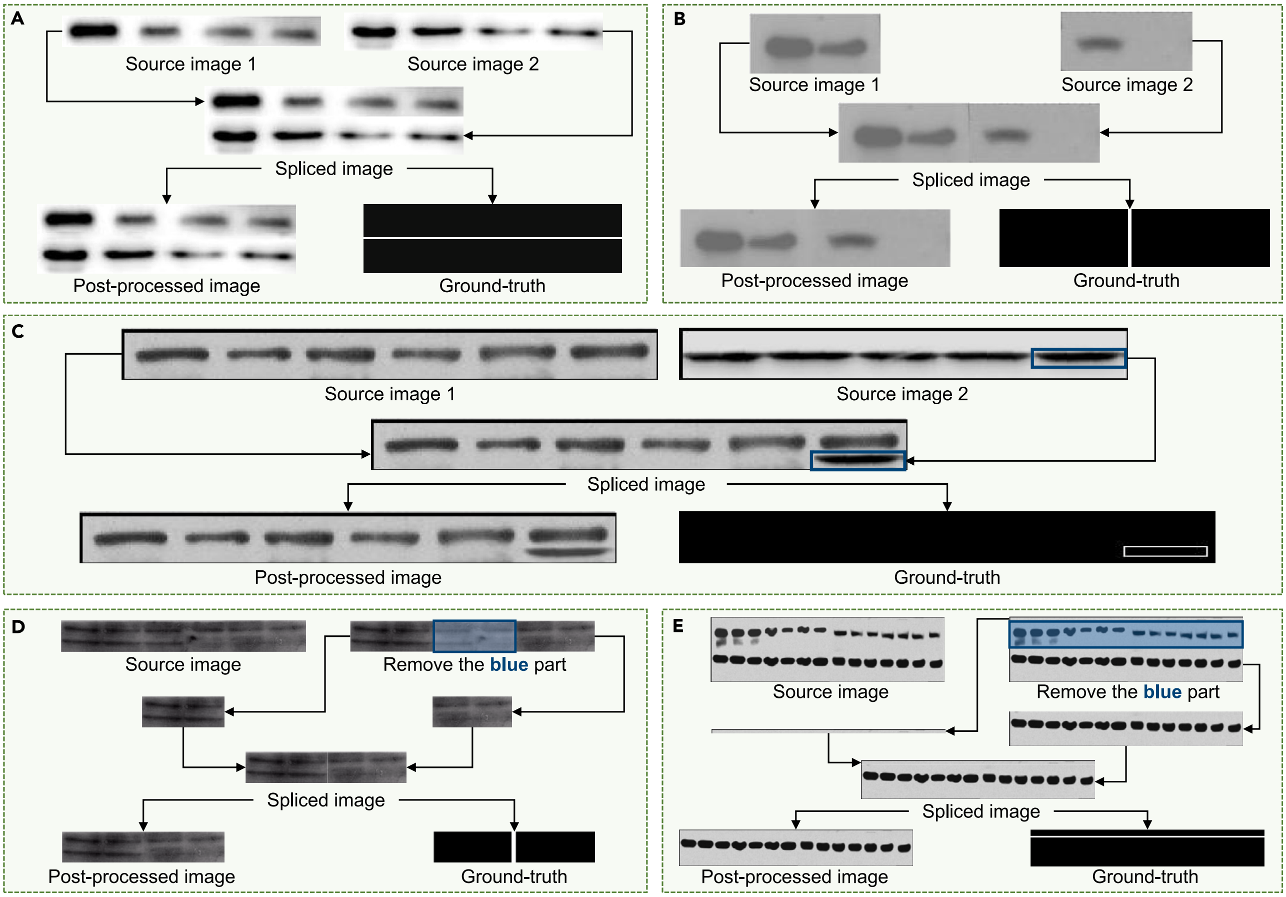

Figure 8. Different splicing approaches for constructing SciSp-H

(A–E) Illustration of five splicing approaches for constructing SciSp-H, including (A) vertical splicing, (B) horizontal splicing, (C) free splicing, (D) vertical removal and splicing, and (E) horizontal removal and splicing.

approach, we created 200 images, all of which were meticulously postprocessed to minimize noticeable visual traces. Because Adobe Photoshop is widely used for image manipulation, we adopted the CC 2019 version to generate spliced images manually. During image splicing and postprocessing, the tools used in Photoshop included content-aware filling, spot-healing brush, brush, cutout, blur, sharpen, and desaturate. All source images in SciSp-H were obtained from images identified as pristine by Biofors.

Adopted datasets for experiments

In addition to our two subdatasets, SciSp-C and SciSp-H, we conducted experiments on two publicly available scientific datasets, Biofors (cut/sharp transition part) and RSIID (splicing part), which are applicable for splicing detection. To prevent interference from false negatives, we randomly sampled pristine images from Biofors for SciSp. We ensured that there was no overlap in the selection of pristine images of Biofors, SciSp-C, and SciSp-H. To avoid biases due to imbalanced data, we sampled pristine images to ensure that their quantity was equal to that of the spliced images. Note that we adopted a training-testing ratio of 7:3 for each dataset. We list the source of all borrowed images presented in this paper in Table S5.

Overall framework of URN

To capture the local inconsistencies caused by splicing, we adopted a CNN as the backbone because of its strong ability to extract local details. As illustrated in Figure 9, our model comprises two stages: estimating the pixel-level uncertainty maps and performing prediction refinement. Given an RGB image $\boldsymbol{x}_i \in \mathbb{R}^{H \times W \times 3}$ as the input, the proposed stage 1 network forwards the encoder-decoder to form the binary coarse mask $\boldsymbol{y}_m \in \{0,1\}^{H \times W}$ and uncertainty map $\boldsymbol{y}_u \in \mathbb{R}^{H \times W}$. Then, the stage 2 network takes $\boldsymbol{y}_m$ and $\boldsymbol{x}_i$ as joint inputs, and $\boldsymbol{y}_u$ is integrated into all the encoder and decoder blocks, ultimately predicting a fine mask $\boldsymbol{y}_v \in \{0,1\}^{H \times W}$. Each encoder unit reduces the height and width of its input by half, whereas each decoder unit doubles them.

The development of neural networks (NNs) is an intuitive solution for distinguishing between spliced and pristine regions. However, traditional NNs cannot accurately estimate the uncertainty in detection results. Additionally, when dealing with images with disruptive factors or out-of-distribution data, they tend to provide incorrect predictions with overconfidence. Therefore, we integrate MCD layers to identify weakly predicted pixels with high uncertainty. Specifically, following Bayesian SegNet,[35] we integrate an MCD layer after each encoder and decoder unit. During inference, the dropouts remain active, enabling the sampling of multiple predictions $\mathcal{S} = \{\boldsymbol{y}_s^1, \boldsymbol{y}_s^2, \boldsymbol{y}_s^3, \cdots, \boldsymbol{y}_s^{n_s}\}$, where n_s is the total number of samples. We take the average value of the samples in $\mathcal{S}$ as the coarse mask $\boldsymbol{y}_m = \sum_{i=1}^{n} \boldsymbol{y}_s^i / n_s$ and use their normalized variance to represent the uncertainty map $\boldsymbol{y}_u = \text{sigmoid}\left(\sqrt{\sum_{i=1}^{n_s} (\boldsymbol{y}_s^i - \boldsymbol{y}_m)^2 / (n_s - 1)}\right)$.

Uncertainty-guided refinement

Uncertainty can accurately reflect confidence in the prediction results of deep-learning-based methods and has been widely applied in computer vision tasks such as image segmentation[49,50] and image classification.[51] The success of uncertainty learning in other fields has promoted its application in image forensic tasks, such as JPEG artifacts,[52] resampling artifacts,[53] and evaluating the confidence of natural image forgery detection results.[26]

Motivated by their work, we introduce uncertainty estimation for robust splicing feature representation learning. In contrast to the aforementioned

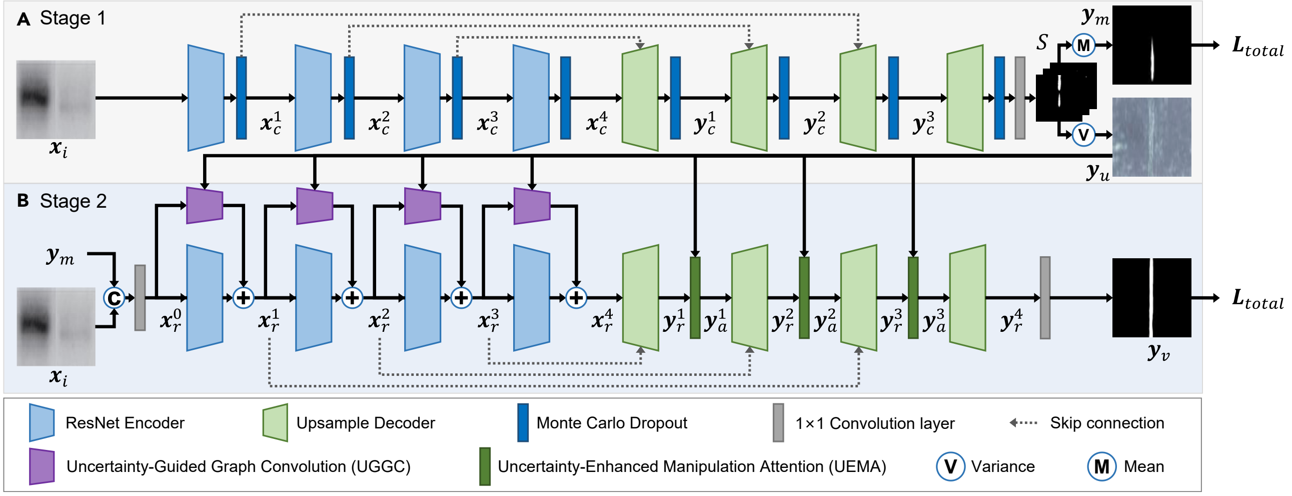

Figure 9. Detailed structure of the proposed URN

uncertainty-integrated methods, we make full use of uncertainty to recognize and refine the uncertain predicted parts affected by disruptive factors. As shown in Figure 9, we propose UGGC and UEMA modules for the stage 2 network. They can leverage uncertainty and refine features during the encoding and decoding stages.

UGGC

Self-attention techniques have been widely studied in media forensics,[5,16,23,26,54,55] leveraging global information to distinguish manipulated from pristine regions.[17] However, the prevalence of disruptive factors in scientific images complicates this task, potentially causing erroneous predictions due to the transmission of unreliable information. This issue remains unresolved, even with the recent k-nearest neighbor (kNN) connection strategies (see Figure 10b).[45] To address this, we propose a UGGC module (see Figure 11), which forms a locally directed weighted graph from feature maps, explicitly guiding the interactions among regions. As seen in Figures 10A–10C, differing from global and kNN connections, we employ an uncertainty-guided connection (UGC) strategy to control information flow direction and intensity between regions. In UGGC, we add local connection constraints to UGC, denoted as local UGC, to help capture subtle differences, such as texture and artifact inconsistencies.

In the specific pipeline of UGGC, we first divide the input feature maps $\boldsymbol{x}_r^k$ and the uncertainty map $\boldsymbol{y}_u$ into patches of size $n_p \times n_p$, where $k \in \{1,2,3,4\}$. To explicitly control the information flow between regions, we construct a locally directed weighted graph $\mathcal{G}^k(\mathcal{N}^k, \mathbf{A}^k)$ using the proposed local UGC strategy. The node set $\mathcal{N}^k$ contains $H_n^k \times W_n^k = H^k/n_p \times W^k/n_p$ nodes, and each node represents a corresponding patch of $\boldsymbol{x}_r^k$. The feature value of each node is the average pooled feature value of the corresponding patch. The number of channels in each node is consistent with $\boldsymbol{x}_r^k$. We use $\boldsymbol{b}_0^k \in \mathbb{R}^{(H_n^k \times W_n^k)\times C^k}$ to represent the initial node features of $\mathcal{N}^k$.

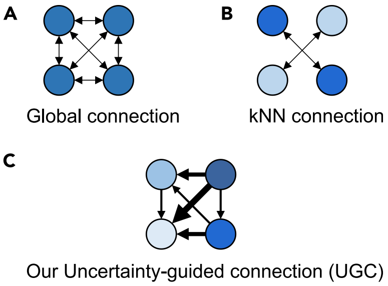

Figure 10. Different edge connection strategies for graph construction

(A–C) Illustration of different edge connection strategies, including (A) global connection, (B) k-nearest neighbor connection, and (C) our uncertainty-guided connection (UGC).

To construct the edge set $\mathbf{A}^k$, we first calculate the initial directed weighted adjacency matrix $\mathbf{E}^k$. The edge weight between nodes $\mathcal{N}_i^k$ and $\mathcal{N}_j^k$ is calculated using Equation 1:

$$\mathbf{E}_{i,j}^k = \begin{cases} \boldsymbol{y}_u^k(j) - \boldsymbol{y}_u^k(i), & \boldsymbol{y}_u^k(i) < \boldsymbol{y}_u^k(j), \\ 0, & \boldsymbol{y}_u^k(i) \geq \boldsymbol{y}_u^k(j), \end{cases} \tag{Equation 1}$$

where $\boldsymbol{y}_u^k(\cdot)$ represents the average uncertainty of the pixels within the corresponding patch in UGGC. As Equation 2 shows, $\mathbf{A}^k$ is derived by adding a local constraint to $\mathbf{E}^k$:

$$\mathbf{A}_{i,j}^k = \begin{cases} \mathbf{E}_{i,j}^k, & |i-j| < 1 \text{ and } |r(i) - r(j)| < 1, \\ 0, & \text{otherwise}, \end{cases} \tag{Equation 2}$$

where $r(\cdot)$ denotes the row index of the input node. Following the construction of $\mathcal{G}_k$, we apply three serially connected graph convolution networks (GCNs).[56] Under local UGC constraints, GCNs can regulate the information transmitted from confident to uncertain nodes. The specific calculation process is as follows:

$$\boldsymbol{b}_\ell^k = \tilde{\mathbf{D}}^{k^{-\frac{1}{2}}} \tilde{\mathbf{A}}^k \tilde{\mathbf{D}}^{k^{-\frac{1}{2}}} \boldsymbol{b}_{\ell-1}^k \mathbf{T}_\ell, \tag{Equation 3}$$

where $\boldsymbol{b}_\ell^k \in \mathbb{R}^{(H_n^k \times W_n^k)\times C_\ell^k}$, $\ell = 1,2,3$ is the input feature map, $\tilde{\mathbf{A}}^k = \mathbf{A}^k + \mathbf{I}$ is the adjacency matrix added with an identity matrix $\mathbf{I}$, $\tilde{\mathbf{D}}^k$ denotes the diagonal node degree matrix of $\tilde{\mathbf{A}}^k$, and $\mathbf{T}_\ell \in \mathbb{R}^{C_{\ell-1}^k \times C_\ell^k}$ is a learnable weight matrix of the ℓ-th GCN. To ensure that the size and number of channels of the output features from UGGC match those of the corresponding ResNet encoder, we let $H_n^k = H_{k+1}$, $W_n^k = W_{k+1}$, $C_1^k = C^k$, and $C_2^k = C_3^k = C^{k+1}$.

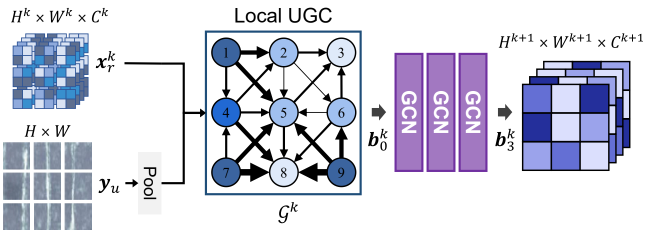

Figure 11. Framework of the proposed UGGC based on our local UGC

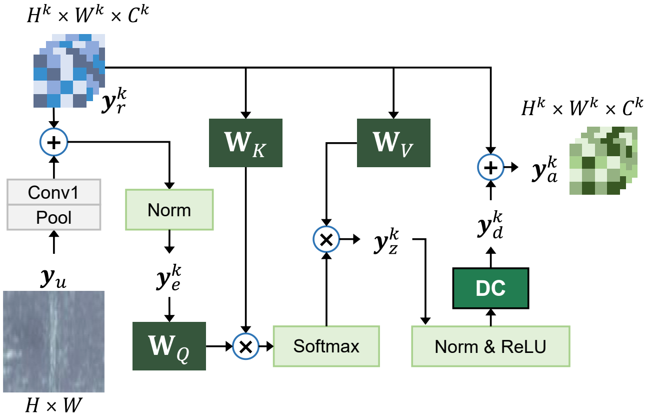

Figure 12. Framework of our UEMA

"DC" is short for a depth-wise convolution layer.

UEMA

In the decoding phase, our goals are to direct more attention toward uncertain regions and align the features of spliced areas while ensuring their distinction from pristine areas. To achieve these two goals simultaneously, we propose a UEMA module (see Figure 12) that fully leverages the uncertainty map and robust features refined by UGGC to improve the performance of the network further. Due to the robust feature representation extracted by UGGC, global correlation computations in UEMA during the decoding phase do not mislead the learning process. First, we use uncertainty maps to enhance the feature maps and perform cross-attention on the feature maps before and after the enhancement. We then adopt depth-wise convolution layers to adaptively adjust the weight of each channel to assist in refining.

Taking $\boldsymbol{y}_r^{(k)}$ and $\boldsymbol{y}_u$ as inputs to UEMA, we sequentially perform average pooling and a convolution layer with 1×1 kernels, making the size of $\boldsymbol{y}_u$ consistent with $\boldsymbol{y}_r^k$. The outputs are given by Equation 4:

$$\boldsymbol{y}_e^k = \mathrm{BN}\left(\boldsymbol{y}_r^k + \mathrm{Conv1}\left(\mathrm{AveragePool}\left(\boldsymbol{y}_u\right)\right)\right), \qquad \text{(Equation 4)}$$

where BN(·) is the batch normalization used to normalize the feature map enhanced by the uncertainty map.

$$\boldsymbol{y}_z^k = \mathrm{Softmax}\left(\frac{\mathbf{W}_Q^k\left(\boldsymbol{y}_e^k\right)\mathbf{W}_K^k\left(\boldsymbol{y}_r^k\right)^\top}{\sqrt{d^k}}\right)\mathbf{W}_V^k\left(\boldsymbol{y}_r^k\right), \qquad \text{(Equation 5)}$$

where $\mathbf{W}_Q^k(\cdot)$, $\mathbf{W}_K^k(\cdot)$, and $\mathbf{W}_V^k(\cdot) \in \mathbb{R}^{d^k \times d^k}$ are the linear projections corresponding to the query, key, and value, respectively, where $d^k = H^k \times W^k$ denotes the number of pixels in a single feature map. Using $\boldsymbol{y}_e^k$ as the query, UEMA can be instructed to focus its attention on uncertain regions and regions suspected of splicing. Additionally, $\boldsymbol{y}_r^k$ serves as the key and value, thereby providing an original basis for global correlation computing. Finally, acknowledging the different contributions from various channels toward the rectification of weakly predicted regions, we incorporated three cascaded depth-wise convolution layers DC(·) (with kernel sizes of 3 × 3)[46] to further refine $\boldsymbol{y}_v^k$, enabling adaptive adjustment of the weights among the channel dimensions.

$$\boldsymbol{y}_a^k = \boldsymbol{y}_z^k + \mathrm{DC}\left(\mathrm{BN}\left(\mathrm{ReLU}\left(\boldsymbol{y}_v^k\right)\right)\right). \qquad \text{(Equation 6)}$$

Loss functions

To simultaneously enhance the pixel- and image-level detection capabilities, we followed previous works[5] by combining classification and segmentation losses ($\mathcal{L}_{cls}$ and $\mathcal{L}_{seg}$, respectively). For $\mathcal{L}_{seg}$, we employed a combination of binary cross-entropy and Dice loss in each stage. The proportion of background pixels far exceeds that of the spliced pixels; thus, we adopted Dice loss due to its capability to deal with pixel imbalance.[5,23] For $\mathcal{L}_{cls}$, we supervise the maximum value of the predicted binary mask $\boldsymbol{y}_v$ to suppress false alarms. The loss functions are formulated as follows:

$$\mathcal{L}_{seg} = \gamma_1 \cdot \mathcal{L}_{bce}(\boldsymbol{y}_v, \boldsymbol{y}) + \gamma_2 \cdot \mathcal{L}_{dice}(\boldsymbol{y}_v, \boldsymbol{y}), \qquad \text{(Equation 7)}$$

$$\mathcal{L}_{cls} = \gamma_3 \cdot \mathcal{L}_{bce}(\boldsymbol{y}_b, \boldsymbol{y}_{cls}), \qquad \text{(Equation 8)}$$

$$\mathcal{L}_{total} = \mathcal{L}_{seg} + \mathcal{L}_{cls}, \qquad \text{(Equation 9)}$$

where $\boldsymbol{y} \in \{0,1\}^{H \times W}$ represents the pixel-level ground truth, with 1 indicating a spliced pixel and 0 indicating a pristine pixel. The variable $\boldsymbol{y}_{cls} \in \{0,1\}$ denotes the image-level ground truth and is defined as $\boldsymbol{y}_{cls} = \max_{\text{pixel}}(\boldsymbol{y})$. This means that $\boldsymbol{y}_{cls}$ is 1 if any pixel in the image $\boldsymbol{y}$ is spliced; otherwise, it is 0. Following previous studies,[5,26] we use $\boldsymbol{y}_b = \max_{\text{pixel}}(\boldsymbol{y}_v)$ to represent image-level prediction, where $\boldsymbol{y}_v$ denotes pixel-level prediction (this prediction is formulated as $\boldsymbol{y}_s$ in stage 1). The empirical parameters γ_1, γ_2, and γ_3 are used to trade off each loss function. Note that $\mathcal{L}_{total}$ is used in both stages of the URN.

Implementation details

We implemented UGNet based on PyTorch 1.7.1. We trained UGNet on a single RTX 3090 GPU for 200 epochs and optimized it using Adam with an initial learning rate of 1×10^{-4}. We set the batch size to eight and resized all images and the corresponding ground-truth labels to 256 × 256. In the stage 1 network, we sample $n_s = 5$ times to estimate the uncertainty maps $\boldsymbol{y}_u$ and predict the coarse masks $\boldsymbol{y}_m$. To improve the training efficiency of the URN and ensure the fairness of the ablation study, we first trained the stage 1 network and then froze its weights during the training phase of the stage 2 network. The entire two-stage training process of URN is shown in the algorithm in Box 1. Additionally, we used the weight of stage 1 to initialize stage 2 for better training efficiency. The decoder structure is shown in Figure 13. The patch size n_p is set to 2. In the URN, the sizes of each intermediate feature map denoted as (H^1, W^1, C^1), (H^2, W^2, C^2), (H^3, W^3, C^3), and (H^4, W^4, C^4) are $(128, 128, 64)$, $(64, 64, 256)$, $(32, 32, 512)$, and $(16, 16, 1{,}024)$, respectively. For the last UEMA module, we downsampled the input feature maps and upsampled the output ones (by 2 ×) due to the limited GPU memory. In Equations 7 and 8, γ_1, γ_2, and γ_3 are 0.7, 0.3, and 0.35, respectively.

Details of evaluation metrics

F1 focuses more on the number of correctly predicted spliced pixels, whereas MCC is frequently used to evaluate the performance of images with imbalanced pixels. MCC produces a high score only if the majority of the predicted negative and positive pixels are correct. These two pixel-level metrics are formulated as follows:

$$\mathrm{F1} = \frac{2 \cdot \mathrm{TP}}{2 \cdot \mathrm{TP} + \mathrm{FP} + \mathrm{FN}}, \qquad \text{(Equation 10)}$$

$$\mathrm{MCC} = \frac{\mathrm{TP} \cdot \mathrm{TN} - \mathrm{FP} \cdot \mathrm{FN}}{\sqrt{(\mathrm{TP}+\mathrm{FP}) \cdot (\mathrm{TP}+\mathrm{FN}) \cdot (\mathrm{TN}+\mathrm{FP}) \cdot (\mathrm{TN}+\mathrm{FN})}}, \qquad \text{(Equation 11)}$$

where TP, TN, FP, and FN denote the numbers of true positive, true negative, false positive, and false negative pixel-level predictions, respectively. The ratio of spliced images to pristine images in our dataset protocols is set to 1:1; hence, we adopt AUC and Acc, which are suitable for evaluating the performance when the categories of datasets are balanced. AUC focuses on the comprehensive performance under different decision thresholds, whereas Acc excels in assessing performance under a fixed threshold (default: 0.5). Acc is formulated as follows:

$$\mathrm{Acc} = \frac{\mathrm{TP}_{cls} + \mathrm{TN}_{cls}}{\mathrm{TP}_{cls} + \mathrm{TN}_{cls} + \mathrm{FP}_{cls} + \mathrm{FN}_{cls}}, \qquad \text{(Equation 12)}$$

where TP_{cls}, TN_{cls}, FP_{cls}, and FN_{cls} denote the numbers of images with true positive, true negative, false positive, and false negative image-level predictions, respectively. We set the default decision thresholds of F1, MCC, and Acc to 0.5 for the following reasons: (1) in real-world applications, determining an optimal decision threshold for unseen testing data seems impractical and (2) setting the same decision threshold for comparison is fair.

Box 1. Training process of URN

Input: stage 1 network $\mathcal{F}_1$, stage 2 network $\mathcal{F}_2$, learning rate ℓ_r, number of samples for MCD n_s, number of training epochs e, input image $\boldsymbol{x}_i$, pixel-level ground-truth label $\boldsymbol{y}_{seg}$, image-level ground-truth label $\boldsymbol{y}_{cls}$, and weights of each loss function, i.e., λ_1, λ_2, and λ_3.

Output: optimized stage 1 network weight $\theta_{\mathcal{F}_1}$ and optimized stage 2 network $\theta_{\mathcal{F}_2}$.

```
# Stage 1: optimize the stage 1 network $\mathcal{F}_1$
for $i \leftarrow 1$ to $e$ do
  $\boldsymbol{y}_s, \boldsymbol{y}_b \leftarrow \mathcal{F}_1(\boldsymbol{x})$;
  $\mathcal{L}_{seg} = \lambda_1 \cdot \mathcal{L}_{bce}(\boldsymbol{y}_s, \boldsymbol{y}_{seg}) + \lambda_2 \cdot \mathcal{L}_{dice}(\boldsymbol{y}_s, \boldsymbol{y}_{seg})$;
  $\mathcal{L}_{cls} = \lambda_3 \cdot \mathcal{L}_{bce}(\boldsymbol{y}_b, \boldsymbol{y}_{cls})$;
  $\mathcal{L}_{total} = \mathcal{L}_{seg} + \mathcal{L}_{cls}$;
  $\theta_{\mathcal{F}_1} \leftarrow \theta_{\mathcal{F}_1} - \ell_r \cdot \nabla_{\theta_{\mathcal{F}_1}}(\mathcal{L}_{total})$;
end
# Stage 2: optimize the stage 2 network $\mathcal{F}_2$
for $i \leftarrow 1$ to $e$ do
  $S \leftarrow \varnothing$
  # Use Monte Carlo sampling to estimate the uncertainty
  for $j \leftarrow 1$ to $n_s$ do
    $\boldsymbol{y}_s, \boldsymbol{y}_b \leftarrow \mathcal{F}_1(\boldsymbol{x})$
    $\mathcal{S} \leftarrow S \cup \boldsymbol{y}_s$
  end
  $\boldsymbol{y}_m = \mathrm{mean}(\mathcal{S})$
  $\boldsymbol{y}_u = \mathrm{variance}(\mathcal{S})$
  # Use the uncertainty to perform refinement
  $\boldsymbol{y}_v, \boldsymbol{y}_b \leftarrow \mathcal{F}_2(\boldsymbol{x}, \boldsymbol{y}_m)$
  $\mathcal{L}_{seg} = \lambda_1 \cdot \mathcal{L}_{bce}(\boldsymbol{y}_v, \boldsymbol{y}_{seg}) + \lambda_2 \cdot \mathcal{L}_{dice}(\boldsymbol{y}_v, \boldsymbol{y}_{seg})$
  $\mathcal{L}_{cls} = \lambda_3 \cdot \mathcal{L}_{bce}(\boldsymbol{y}_b, \boldsymbol{y}_{cls})$
  $\mathcal{L}_{total} = \mathcal{L}_{seg} + \mathcal{L}_{cls}$
  $\theta_{\mathcal{F}_2} \leftarrow \theta_{\mathcal{F}_2} - \ell_r \cdot \nabla_{\theta_{\mathcal{F}_2}}(\mathcal{L}_{total})$
end
```

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.patter.2024.101038.

ACKNOWLEDGMENTS

Support funding was provided by the National Natural Science Foundation of China (NSFC) under grant no. 62272022. This work was also supported by the National Key Research and Development Program of China under grants 210YBXM2024106007 and 2022YFB3207700. Additional support came from the NSFC under grants U22A2009 and 62306061.

AUTHOR CONTRIBUTIONS

Conceptualization, X.L. and S.W.; writing – original draft, X.L.; methodology, X.L., Y.J., Z.Y., and S.W.; data curation, X.L. and H.W.; investigation, X.L., H.W., and Y.L.; formal analysis, W.T., H.W., Y.L., and X.L.; writing – review & editing, Z.Y., Y.J., and H.W.; visualization, Y.L. and X.L.; supervision, Z.Y.; funding acquisition, W.T. and S.W.

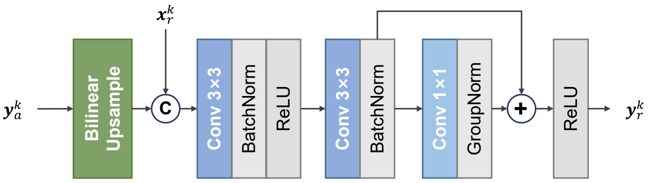

Figure 13. Structure of the decoder block in URN

DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: April 19, 2024
Revised: May 16, 2024
Accepted: July 8, 2024
Published: August 8, 2024

REFERENCES

1. Rossner, M., and Yamada, K.M. (2004). What's in a picture? the temptation of image manipulation. J. Cell Biol. *166*, 11–15. https://doi.org/10.1083/jcb.200406019.
2. Bik, E.M., Casadevall, A., and Fang, F.C. (2016). The prevalence of inappropriate image duplication in biomedical research publications. mBio *7*, e00809-16. https://doi.org/10.1128/mbio.00809-16.
3. Miura, K., and Nørrelykke, S.F. (2021). Reproducible image handling and analysis. EMBO J. *40*, e105889. https://doi.org/10.15252/embj.2020105889.
4. Bucci, E.M. (2018). Automatic detection of image manipulations in the biomedical literature. Cell Death Dis. *9*, 400–409. https://doi.org/10.1038/s41419-018-0430-3.
5. Dong, C., Chen, X., Hu, R., Cao, J., and Li, X. (2023). Mvss-net: Multi-view multi-scale supervised networks for image manipulation detection. IEEE Trans. Pattern Anal. Mach. Intell. *45*, 3539–3553. https://doi.org/10.1109/TPAMI.2022.3180556.
6. Sabir, E., Nandi, S., AbdAlmageed, W., and Natarajan, P. (2021). Biofors: A Large Biomedical Image Forensics Dataset. In IEEE/CVF International

Conference on Computer Vision, pp. 10943–10953. https://doi.org/10.1109/ICCV48922.2021.01078.

7. Cardenuto, J.P., and Rocha, A. (2022). Benchmarking scientific image forgery detectors. Sci. Eng. Ethics *28*, 35. https://doi.org/10.1007/s11948-022-00391-4.

8. Koker, T.E., Chintapalli, S.S., Wang, S., Talbot, B.A., Wainstock, D., Cicconet, M., and Walsh, M.C. (2020). On identification and retrieval of near-duplicate biological images: a new dataset and protocol. In International Conference on Pattern Recognition, pp. 3114–3121. https://doi.org/10.1109/ICPR48806.2021.9412849.

9. Moreira, D., Cardenuto, J.P., Shao, R., Baireddy, S., Cozzolino, D., Gragnaniello, D., Abd-Almageed, W., Bestagini, P., Tubaro, S., Rocha, A., et al. (2022). Sila: a system for scientific image analysis. Sci. Rep. *12*, 18306. https://doi.org/10.1038/s41598-022-21535-3.

10. Sabir, E., Nandi, S., AbdAlmageed, W., and Natarajan, P. (2022). Monet: Multi-scale overlap network for duplication detection in biomedical images. In IEEE International Conference on Image Processing, pp. 3793–3797. https://doi.org/10.1109/ICIP46576.2022.9897213.

11. Gu, J., Wang, X., Li, C., Zhao, J., Fu, W., Liang, G., and Qiu, J. (2022). AI-enabled image fraud in scientific publications. Patterns *3*, 100511. https://doi.org/10.1016/j.patter.2022.100511.

12. Wang, L., Zhou, L., Yang, W., and Yu, R. (2022a). Deepfakes: A new threat to image fabrication in scientific publications? Patterns *3*, 100509. https://doi.org/10.1016/j.patter.2022.100509.

13. Verdoliva, L. (2020). Media forensics and deepfakes: An overview. IEEE J. Sel. Top. Signal Process. *14*, 910–932. https://doi.org/10.1109/JSTSP.2020.3002101.

14. Zhou, P., Chen, B., Han, X., Najibi, M., Shrivastava, A., Lim, S., and Davis, L. (2020). Generate, segment, and refine: Towards generic manipulation segmentation. In AAAI Conference on Artificial Intelligence, pp. 13058–13065. https://doi.org/10.1609/aaai.v34i07.7007.

15. Bi, X., Wei, Y., Xiao, B., and Li, W. (2019). Rru-net: The ringed residual u-net for image splicing forgery detection. In IEEE/CVF Computer Vision and Pattern Recognition Conference Workshop, pp. 30–39. https://doi.org/10.1109/CVPRW.2019.00010.

16. Hao, J., Zhang, Z., Yang, S., Xie, D., and Pu, S. (2021). Transforensics: Image Forgery Localization with Dense Self-Attention. In IEEE/CVF International Conference on Computer Vision, pp. 15035–15044. https://doi.org/10.1109/ICCV48922.2021.01478.

17. Liu, X., Liu, Y., Chen, J., and Liu, X. (2022). Pscc-net: Progressive spatio-channel correlation network for image manipulation detection and localization. IEEE Trans. Circuits Syst. Video Technol. *32*, 7505–7517. https://doi.org/10.1109/TCSVT.2022.3189545.

18. Kwon, M., Yu, I., Nam, S., and Lee, H. (2021). Cat-net: Compression artifact tracing network for detection and localization of image splicing. In IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 375–384. https://doi.org/10.1109/WACV48630.2021.00042.

19. Kwon, M., Nam, S., Yu, I., Lee, H., and Kim, C. (2022). Learning JPEG compression artifacts for image manipulation detection and localization. Int. J. Comput. Vis. *130*, 1875–1895. https://doi.org/10.1007/s11263-022-01617-5.

20. Wang, J., Wu, Z., Chen, J., Han, X., Shrivastava, A., Lim, S., and Jiang, Y. (2022b). Objectformer for image manipulation detection and localization. In IEEE/CVF Computer Vision and Pattern Recognition Conference, pp. 2354–2363. https://doi.org/10.1109/CVPR52688.2022.00240.

21. Zhou, P., Han, X., Morariu, V.I., and Davis, L.S. (2018). Learning rich features for image manipulation detection. In IEEE/CVF Computer Vision and Pattern Recognition Conference, pp. 1053–1061. https://doi.org/10.1109/CVPR.2018.00116.

22. Wu, Y., AbdAlmageed, W., and Natarajan, P. (2019). Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In IEEE/CVF Computer Vision and Pattern Recognition Conference, pp. 9543–9552. https://doi.org/10.1109/CVPR.2019.00977.

23. Lin, X., Wang, S., Deng, J., Fu, Y., Bai, X., Chen, X., Qu, X., and Tang, W. (2023). Image manipulation detection by multiple tampering traces and edge artifact enhancement. Pattern Recognit *133*, 109026. https://doi.org/10.1016/j.patcog.2022.109026.

24. Wang, H., Deng, J., Lin, X., Tang, W., and Wang, S. (2023). Cds-net: Cooperative dual-stream network for image manipulation detection. Pattern Recognit. Lett. *176*, 167–173. https://doi.org/10.1016/j.patrec.2023.11.005.

25. Cozzolino, D., and Verdoliva, L. (2020). Noiseprint: A cnn-based camera model fingerprint. IEEE Trans. Inf. Forensics Secur. *15*, 144–159. https://doi.org/10.1109/TIFS.2019.2916364.

26. Guillaro, F., Cozzolino, D., Sud, A., Dufour, N., and Verdoliva, L. (2023). Trufor: Leveraging all-round clues for trustworthy image forgery detection and localization. In IEEE/CVF Computer Vision and Pattern Recognition Conference, pp. 20606–20615. https://doi.org/10.1109/CVPR52729.2023.01974.

27. Guo, X., Liu, X., Ren, Z., Grosz, S., Masi, I., and Liu, X. (2023). Hierarchical fine-grained image forgery detection and localization. In IEEE/CVF Computer Vision and Pattern Recognition Conference, pp. 3155–3165. https://doi.org/10.1109/CVPR52729.2023.00308.

28. Hu, J., Liao, X., Wang, W., and Qin, Z. (2022). Detecting compressed deepfake videos in social networks using frame-temporality two-stream convolutional network. IEEE Trans. Circuits Syst. Video Technol. *32*, 1089–1102. https://doi.org/10.1109/TCSVT.2021.3074259.

29. Wu, H., Zhou, J., Tian, J., and Liu, J. (2022). Robust image forgery detection over online social network shared images. In IEEE/CVF Computer Vision and Pattern Recognition Conference, pp. 13430–13439. https://doi.org/10.1109/CVPR52688.2022.01308.

30. Zhuang, P., Li, H., Tan, S., Li, B., and Huang, J. (2021). Image tampering localization using a dense fully convolutional network. IEEE Trans. Inf. Forensics Secur. *16*, 2986–2999. https://doi.org/10.1109/TIFS.2021.3070444.

31. Wang, S., Wang, O., Zhang, R., Owens, A., and Efros, A.A. (2019). Detecting photoshopped faces by scripting photoshop. In IEEE/CVF International Conference on Computer Vision, pp. 10071–10080. https://doi.org/10.1109/ICCV.2019.01017.

32. Suvorov, R., Logacheva, E., Mashikhin, A., Remizova, A., Ashukha, A., Silvestrov, A., Kong, N., Goka, H., Park, K., and Lempitsky, V. (2022). Resolution-robust large mask inpainting with fourier convolutions. In IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 3172–3182. https://doi.org/10.1109/WACV51458.2022.00323.

33. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In IEEE/CVF Computer Vision and Pattern Recognition Conference, pp. 10684–10695. https://doi.org/10.1109/CVPR52688.2022.01042.

34. Bi, X., Shang, Y., Liu, B., Xiao, B., Li, W., and Gao, X. (2023). A versatile detection method for various contrast enhancement manipulations. IEEE Trans. Circuits Syst. Video Technol. *33*, 491–504. https://doi.org/10.1109/TCSVT.2022.3204789.

35. Badrinarayanan, V., Kendall, A., and Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. *39*, 2481–2495. https://doi.org/10.1109/TPAMI.2016.2644615.

36. Mandelli, S., Cozzolino, D., Cannas, E.D., Cardenuto, J.P., Moreira, D., Bestagini, P., Scheirer, W.J., Rocha, A., Verdoliva, L., Tubaro, S., and Delp, E.J. (2022). Forensic analysis of synthetically generated western blot images. IEEE Access *10*, 59919–59932. https://doi.org/10.1109/ACCESS.2022.3179116.

37. Chicco, D., Warrens, M.J., and Jurman, G. (2021). The matthews correlation coefficient (MCC) is more informative than cohen's kappa and brier score in binary classification assessment. IEEE Access *9*, 78368–78381. https://doi.org/10.1109/ACCESS.2021.3084050.

38. Wei, Y., Ma, J., Wang, Z., Xiao, B., and Zheng, W. (2022). Image splicing forgery detection by combining synthetic adversarial networks and hybrid dense u-net based on multiple spaces. Int. J. Intell. Syst. *37*, 8291–8308.

39. Shao, H., Cheng, Y., Duh, M., and Lin, C. (2020). Forgery blind inspection for detecting manipulations of gel electrophoresis images. Preprint at arXiv. https://doi.org/10.48550/arXiv.2010.15086.

40. Guan, H., Kozak, M., Robertson, E., Lee, Y., Yates, A.N., Delgado, A., Zhou, D., Kheyrkhah, T., Smith, J., and Fiscus, J.G. (2019). MFC datasets: Large-scale benchmark datasets for media forensic challenge evaluation. In IEEE/CVF Winter Conference on Applications of Computer Vision Workshop, pp. 63–72. https://doi.org/10.1109/WACVW.2019.00018.

41. Wu, H., and Zhou, J. (2022). Iid-net: Image inpainting detection network via neural architecture search and attention. IEEE Trans. Circuits Syst. Video Technol *32*, 1172–1185. https://doi.org/10.1109/TCSVT.2021.3075039.

42. Bertalmío, M., Bertozzi, A.L., and Sapiro, G. (2001). Navier-stokes, fluid dynamics, and image and video inpainting. In IEEE/CVF Computer Vision and Pattern Recognition Conference, pp. 355–362. https://doi.org/10.1109/CVPR.2001.990497.

43. Telea, A.C. (2004). An image inpainting technique based on the fast marching method. J. Graphics, GPU, & Game Tools *9*, 23–34. https://doi.org/10.1080/10867651.2004.10487596.

44. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2020). Grad-cam: Visual explanations from deep networks via gradient-based localization. Int. J. Comput. Vis. *128*, 336–359. https://doi.org/10.1007/s11263-019-01228-7.

45. Han, K., Wang, Y., Guo, J., Tang, Y., and Wu, E. (2022). Vision GNN: an image is worth graph of nodes. In Conference on Neural Information Processing Systems https://dl.acm.org/doi/abs/10.5555/3600270.3600873.

46. Chen, Q., Wu, Q., Wang, J., Hu, Q., Hu, T., Ding, E., Cheng, J., and Wang, J. (2022). Mixformer: Mixing features across windows and dimensions. In IEEE/CVF Computer Vision and Pattern Recognition Conference, pp. 5239–5249. https://doi.org/10.1109/CVPR52688.2022.00518.

47. Lin, X. (2024a). Code for "Exposing Image Splicing Traces in Scientific Publications via Uncertainty-Guided Refinement. Preprint at Zenodo. https://doi.org/10.5281/zenodo.12578993.

48. Lin, X. (2024b). Datasets for "Exposing Image Splicing Traces in Scientific Publications via Uncertainty-Guided Refinement. Preprint at Zenodo. https://doi.org/10.5281/zenodo.10989920.

49. Zheng, E., Yu, Q., Li, R., Shi, P., and Haake, A.R. (2021). A continual learning framework for uncertainty-aware interactive image segmentation. In AAAI Conference on Artificial Intelligence, pp. 6030–6038. https://doi.org/10.1609/aaai.v35i7.16752.

50. Qu, X., Zhou, J., Jiang, J., Wang, W., Wang, H., Wang, S., Tang, W., and Lin, X. (2024). Eh-former: Regional easy-hard-aware transformer for breast lesion segmentation in ultrasound images. Inf. Fusion *109*, 102430. https://doi.org/10.1016/j.inffus.2024.102430.

51. Lin, X., Wang, S., Cai, R., Liu, Y., Fu, Y., Yu, Z., Tang, W., and Kot, A.C. (2024). Suppress and rebalance: Towards generalized multi-modal face anti-spoofing. Preprint at arXiv. https://doi.org/10.48550/arXiv.2402.19298.

52. Lorch, B., Maier, A., and Riess, C. (2020). Reliable JPEG forensics via model uncertainty. In IEEE International Workshop on Information Forensics and Security, pp. 1–6. https://doi.org/10.1109/WIFS49906.2020.9360893.

53. Maier, A., Lorch, B., and Riess, C. (2020). Toward reliable models for authenticating multimedia content: Detecting resampling artifacts with bayesian neural networks. In IEEE International Conference on Image Processing, pp. 1251–1255. https://doi.org/10.1109/ICIP40778.2020.9191121.

54. Hu, X., Zhang, Z., Jiang, Z., Chaudhuri, S., Yang, Z., and Nevatia, R. (2020). SPAN: spatial pyramid attention network for image manipulation localization. European Conference on Computer Vision *12366*, 312–328. https://doi.org/10.1007/978-3-030-58589-1_19.

55. Binh, L.M., and Woo, S.S. (2022). ADD: frequency attention and multi-view based knowledge distillation to detect low-quality compressed deepfake images. In AAAI Conference on Artificial Intelligence, pp. 122–130. https://doi.org/10.1609/aaai.v36i1.19886.

56. Kipf, T.N., and Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In International Conference on Learning Representations https://openreview.net/forum?id=SJU4ayYgl.