Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Image Tampering Region Localization Method Based on Pseudo Label Driver and Dynamic Multi branch Decoding

Yanting Li
Chengyi College, Jimei University
Xiaojian Xue
Chengyi College, Jimei University
Weiyi Lin
Chengyi College, Jimei University
Shouxin He
ShouxinHe63@outlook.com

Xiamen Security Technology Vocational College

Research Article

Keywords: Image tampering localization, pseudo-label learning, dynamic multi-branch decoder, copy-move forgery detection, boundary refinement

Posted Date: September 15th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10703948/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Image Tampering Region Localization Method Based on Pseudo Label Driver and Dynamic Multi branch Decoding

Yanting Li [1], Xiaojian Xue [2],Weiyi Lin [3],Shouxin He* [4]

[1、2、3]Chengyi College, Jimei University, Xiamen Fujian; 361021, China

[4.]Xiamen Security Technology Vocational College, Xiamen, China.

Corresponding author:Shouxin He

Yanting Li's e-mail:YantingLi28@outlook.com

Xiaojian Xue's e-mail:XiaojianXue6@outlook.com

Weiyi Lin's e-mail:WeiyiLin2@outlook.com

Shouxin He's e-mail:ShouxinHe63@outlook.com

Funding

2024 Fujian Province Young and Middle aged Teacher Education Research Project (Technology Category) Project Leader: Li Yanting Project Number: JAT242021 Project Name: Computer Network Security Monitoring and Protection Based on Deep Learning Technology

Author contributions:

Yanting Li: Conceptualization; Methodology; Formal analysis; Investigation; Writing – original draft; Writing – review & editing; Supervision; Project administration; Funding acquisition.

Xiaojian Xue: Software; Validation; Data curation; Visualization; Writing – original draft; Writing – review & editing.

Weiyi Lin: Software; Validation; Data curation; Visualization; Investigation.

Shouxin He: Conceptualization; Methodology; Formal analysis; Investigation; Writing – original draft; Writing – review & editing; Supervision; Project administration.

Conflict of interest:

The author declares that they have no competing interest.

Data availability statement:

The datasets generated during and/or analysed during the current study are available from the corr esponding author on reasonable request.

Ethical approval

Not applicable.

Consent to participate

Not applicable.

Consent to publish

Not applicable.

Abstract： Image tampering localization aims to identify manipulated pixels at a

fine-grained level; however, heterogeneous forensic traces among splicing, inpainting, and copy-move forgeries make this task challenging. To address these issues, this paper proposes a pseudo-label-driven image tampering localization method with a tamper-type-aware dynamic multi-branch decoder. The proposed framework first employs a multi-scale encoder to extract both low-level forensic traces and high-level semantic representations. To explicitly model copy-move forgery, a pseudo-label generation module is introduced to construct source–target correspondence supervision with confidence estimation, enabling weakly supervised learning of duplicated regions. Furthermore, a dynamic multi-branch decoder is designed to adaptively integrate different forensic cues, including boundary inconsistency, texture disturbance, and source–target similarity, according to tampering type and pseudo-label reliability. Extensive experiments demonstrate the effectiveness of the proposed method. In cross-dataset evaluation, the model achieves 75.4% F1 and 93.5% AUC on the COLUMBIA dataset, and 52.7% F1 and 87.2% AUC on the COVER dataset. In intra-dataset testing, it reaches up to 99.0% F1 on COLUMBIA and 98.6% F1 on COVER, demonstrating strong localization accuracy and generalization ability.

Keywords：Image tampering localization; pseudo-label learning; dynamic multi-branch decoder; copy-move forgery detection; boundary refinement

1. Introduction

With the rapid development of digital imaging devices, image editing software, social media platforms, and artificial intelligence generated content, digital images have become one of the most important carriers of visual information. They are widely used in news reporting, judicial evidence, public security, medical diagnosis, remote sensing interpretation, and online communication. The authenticity of digital images is therefore closely related to information credibility, evidence reliability, and social trust. However, modern image manipulation techniques allow users to alter image content with very low technical cost. Operations such as splicing, copy-move, object removal, inpainting, local retouching, and recompression can modify the semantic meaning of an image while preserving a visually realistic appearance. In many practical scenarios, tampered images are difficult to identify by human observation

alone, especially when manipulated regions are small, carefully blended, or further degraded during online transmission. Therefore, automatic image tampering region localization has become an important research topic in digital image forensics [1].

Image tampering region localization is more challenging than image-level forgery detection. Image-level detection only determines whether an image has been manipulated, whereas region localization requires the model to identify tampered pixels and generate an accurate mask of the manipulated area. This pixel-level requirement is significant because it provides interpretable forensic evidence for subsequent analysis, such as determining which object has been inserted, removed, duplicated, or modified. In real forensic applications, localization results can help investigators understand manipulation intention, evaluate the reliability of visual evidence, and support decision-making in legal or security-sensitive scenarios. Consequently, an effective localization method should not only distinguish authentic and tampered images, but also preserve the completeness, boundary accuracy, and structural consistency of manipulated regions [2].

Despite continuous progress, image tampering localization remains difficult due to the diversity of manipulation types and the weakness of forensic traces. Different manipulation operations usually leave different visual and statistical artifacts. Splicing may introduce boundary discontinuities, illumination inconsistency, noise mismatch, or compression inconsistency because the inserted region often comes from another image. Inpainting may create abnormal texture compensation, structural smoothing, or local semantic inconsistency. Copy-move manipulation is more difficult because the copied source region and the pasted target region originate from the same image. As a result, they often share similar color distribution, texture pattern, noise characteristics, and imaging conditions. This intrinsic homology makes copy-move forgery less dependent on external inconsistency cues and more reliant on source–target relationship modeling [3].

Existing localization methods have achieved promising results by introducing handcrafted forensic descriptors, convolutional neural networks, multi-scale feature fusion, frequency-domain analysis, and boundary-aware supervision. However, several limitations still remain. First, many methods formulate image tampering localization as a conventional binary segmentation task and process all manipulation types through a unified decoding

pathway. Such a design may be insufficient because heterogeneous manipulation traces are forced into the same feature representation and prediction process. When splicing, inpainting, and copy-move samples are jointly involved in training, a single decoder may learn averaged responses and fail to capture type-specific forensic evidence. Second, copy-move localization requires the model to discover the relationship between the source and target regions, but most datasets only provide the pasted target mask, while the corresponding source region is not explicitly annotated. Third, small tampered regions are easily overwhelmed by a large number of authentic background pixels. During downsampling and semantic aggregation, weak boundaries and subtle local artifacts may be suppressed, leading to missed detection, incomplete masks, or fragmented localization results. Finally, manipulated images are often compressed, resized, blurred, noised, or re-uploaded through social platforms before being examined. These post-processing operations can weaken forensic traces and reduce model robustness [4].

To address these problems, this paper proposes an image tampering region localization method based on pseudo-label driving and dynamic multi-branch decoding. The proposed framework is designed from the perspective of manipulation heterogeneity. Instead of directly mapping an input image to a single binary mask, the model jointly considers general tampered-region prediction, copy-move source–target modeling, and tamper-type-aware feature decoding. A multi-scale encoder is first used to extract hierarchical features, where shallow layers preserve low-level forensic traces such as texture disturbance, noise inconsistency, and boundary artifacts, while deep layers provide semantic context for distinguishing plausible image content from manipulated regions. This multi-scale representation provides a basis for both small-region localization and type-specific feature enhancement.

For copy-move manipulation, this paper introduces a pseudo-label-driven source–target learning strategy. Since manually annotated source masks are often unavailable, the proposed method uses the tampered-region prior, feature similarity search, morphological exclusion, and confidence estimation to generate weak source–target pseudo-labels. The generated pseudo-label describes the union of the possible source region and the pasted target region, enabling the network to learn copy-move homology explicitly. A confidence score is further

introduced to reduce the negative influence of unreliable pseudo-labels, especially in cases with repetitive background textures, severe post-processing, or small duplicated regions. Through iterative updating, the pseudo-label quality can be gradually refined as the localization branch becomes more stable.

In addition, a tamper-type-aware dynamic multi-branch decoder is constructed to adaptively handle heterogeneous manipulation traces. Different branches are designed to focus on different forensic cues, including source–target similarity, boundary inconsistency, local texture disturbance, and generic tampering evidence. A gating mechanism estimates branch weights according to the tamper-type response and pseudo-label reliability, allowing the decoder to emphasize the most relevant branch for the current input. For example, the source–target branch can contribute more strongly when the input image is likely to contain copy-move forgery and the generated pseudo-label is reliable, whereas boundary-sensitive or texture-sensitive branches can be emphasized for splicing or inpainting cases. Furthermore, a class-balanced and boundary-refined joint loss is designed to address small-region localization. The category-balanced term increases the contribution of minority tampered pixels, the Dice loss enhances region-level completeness, and the boundary refinement loss improves contour alignment.

The main contributions of this paper are summarized as follows. First, a unified image tampering localization framework is proposed by integrating pixel-level tampered-region prediction, copy-move source–target relationship learning, and tamper-type-aware dynamic decoding. Second, a pseudo-label-driven weak supervision mechanism is designed for copy-move forgery, enabling the model to learn source–target homology without requiring complete manual source-region annotations. Third, a dynamic multi-branch decoder is proposed to adaptively fuse different manipulation-sensitive responses, improving the model's ability to handle heterogeneous forensic traces. Fourth, a class-imbalance optimization and boundary refinement strategy is introduced to improve the localization of small tampered regions and enhance mask completeness and boundary consistency.

2. Literature Review

Image tampering detection and localization have evolved from traditional forensic

feature analysis to deep learning based pixel-level prediction. Early passive forensic studies mainly focused on statistical traces generated during image acquisition, geometric transformation, compression, and post-processing. Popescu and Farid investigated resampling traces caused by scaling, rotation, and other geometric operations, showing that interpolation artifacts can reveal local image modifications [5]. This type of method is interpretable because it relies on explicit signal processing assumptions, but its effectiveness may decrease when the manipulated image is further recompressed or smoothed.

Compression inconsistency is another important clue in passive image forensics. Farid studied JPEG ghosts and demonstrated that inconsistent compression histories can expose locally manipulated regions [6]. Compared with resampling-based analysis, JPEG-related methods are more suitable for images that have undergone block-based compression, but they may be less effective when compression traces are weak or uniformly recompressed. Lukas, Fridrich, and Goljan explored sensor pattern noise for digital camera identification, which later became an important basis for detecting local noise inconsistency in forged images [7]. These studies indicate that low-level acquisition traces can provide valuable forensic evidence, but they are often sensitive to post-processing operations.

To further improve localization granularity, Bianchi and Piva analyzed JPEG artifacts at the block level and used compression inconsistency for image forgery localization [8]. Their work suggests that block-wise analysis can support more detailed region localization than image-level detection. Ferrara, Bianchi, De Rosa, and Piva focused on color filter array artifacts and showed that fine-grained inconsistencies in the imaging pipeline could indicate tampered regions [9]. Overall, traditional forensic methods provide clear physical or statistical explanations, but their dependence on specific traces limits their generalization ability under complex real-world manipulation pipelines.

Copy-move forgery detection has been another important branch of image forensics. Unlike splicing, copy-move manipulation duplicates content from the same image, making source and target regions visually and statistically similar. Amerini, Ballan, Caldelli, Del Bimbo, and Serra used SIFT features to detect copy-move attacks and estimate geometric transformations, making keypoint matching a representative strategy for duplicated-region detection [10]. This method is robust to certain transformations such as rotation and scaling,

but sparse keypoints may be insufficient in smooth or low-texture areas.

A broader comparison of copy-move methods was provided by Christlein, Riess, Jordan, Riess, and Angelopoulou, who evaluated popular copy-move forgery detection approaches and pointed out the differences between block-based and keypoint-based strategies [11]. Block-based methods can produce denser matching results, while keypoint-based methods usually have better resistance to geometric transformations. However, both types of methods may suffer from false matches in repetitive textures. To improve dense matching, Cozzolino, Poggi, and Verdoliva proposed an efficient dense-field copy-move forgery detection method, which enhanced localization completeness compared with sparse matching approaches [12].

Several studies further attempted to improve the spatial coherence and geometric consistency of copy-move localization. Li, Li, Yang, and Sun introduced a segmentation-based copy-move forgery detection scheme to reduce false matches and improve region consistency [13]. Ardizzone, Bruno, and Mazzola used keypoint triangle matching to strengthen geometric constraints during duplicated-region detection [14]. These methods effectively characterize intra-image similarity, but many of them are designed specifically for copy-move forgery. They are not easily generalized to a unified multi-type tampering localization framework. Moreover, traditional matching-based methods usually focus on finding duplicated regions, but they do not always distinguish the source region from the tampered target region.

With the development of convolutional neural networks, image tampering localization has gradually shifted toward end-to-end dense prediction. Salloum, Ren, and Kuo proposed a multi-task fully convolutional network for image splicing localization, where the network jointly learned manipulation masks and boundary information [15]. This work shows that boundary supervision is useful because many tampering artifacts are concentrated around transition regions. Bappy, Simons, Nataraj, Manjunath, and Roy-Chowdhury developed a hybrid LSTM and encoder–decoder architecture for image forgery detection, combining resampling features with contextual modeling to capture manipulation evidence across larger spatial regions [16].

Context modeling and adversarial learning have also been introduced into manipulation localization. Liu, Zhu, Zhao, and Cao proposed an adversarial learning framework based on

atrous convolution for constrained image splicing detection and localization, improving the ability to exploit contextual inconsistency [17]. Compared with traditional handcrafted features, such deep models can learn more complex manipulation patterns from data. However, they still largely follow a fixed prediction pipeline and may not sufficiently distinguish manipulation-specific cues.

For copy-move forgery, Zhong and Pun proposed Dense-InceptionNet for end-to-end detection, using dense connections and inception-style structures to extract multi-scale duplicated-region features [18]. This work demonstrates that deep networks can replace traditional handcrafted matching to some extent. Weng, Zhu, Zhang, and Zhang further proposed UCM-Net, a U-Net-like tampered-region-related framework for copy-move forgery detection, showing the effectiveness of encoder–decoder structures in duplicated-region localization [19]. Nevertheless, many copy-move deep models still focus on duplicated-region detection itself and do not fully solve the problem of source–target supervision deficiency.

Multi-scale feature fusion has been widely studied because tampered regions may appear at different sizes and contain both low-level artifacts and high-level semantic inconsistency. Li, Luo, Qiu, and Huang proposed to integrate tampering possibility maps from multiple forensic cues, demonstrating that complementary evidence can improve forgery localization [20]. Compared with single-cue methods, multi-cue integration is more robust to diverse manipulations. However, when different cues are simply fused without type-aware selection, redundant or unreliable responses may also introduce false positives.

To improve pixel-level localization, Li, Pei, Zhang, and Qin proposed image manipulation localization using multi-scale feature fusion and adaptive edge supervision [21]. Their method jointly exploits multi-level representations and boundary information, which is beneficial for locating small or irregular manipulated regions. Dong, Chen, Hu, Cao, and Li proposed MVSS-Net, which uses multi-view and multi-scale supervision to jointly learn boundary artifacts and noise-view features [22]. The significance of MVSS-Net lies in its attempt to learn semantic-agnostic manipulation features that generalize better across datasets. However, these methods still mainly rely on a unified decoding process and do not explicitly model different decoding paths for different manipulation types.

Frequency-domain and compression-domain representations have also been explored in

recent image manipulation localization. Kwon, Nam, Yu, Lee, and Kim proposed CAT-Net, which learns JPEG compression artifacts from DCT-domain representations and combines acquisition and compression traces for manipulation localization [23]. This approach is effective for detecting compression-related inconsistencies, especially when manipulation changes local JPEG histories. Gu, Nam, and Lee proposed FBI-Net, which introduces frequency decomposition, multitask learning, and self-attention to strengthen frequency-domain forgery localization [24]. These studies show that effective localization requires the integration of spatial, boundary, noise, compression-related, and frequency-domain information. Nevertheless, frequency-domain features may be weakened by repeated compression or quality degradation, and they still need to be combined with adaptive region-level decoding.

Recent research has also explored source–target distinction in copy-move forgery. Chen, Tan, Coatrieux, Zheng, and Shi proposed a serial image copy-move forgery localization scheme with source/target distinguishment, emphasizing that separating copied source regions from pasted target regions is important for forensic interpretation [25]. This direction is highly relevant because conventional copy-move localization often identifies duplicated regions but does not clearly indicate which region has been tampered with. Source–target distinguishment provides richer forensic evidence, but it usually requires more specific supervision or carefully designed matching processes. In many public datasets, only the forged target mask is available, and complete source-region annotation is missing. Therefore, learning source–target relationships under weak supervision remains an important problem.

From the perspective of network design, dynamic and adaptive decoding is still insufficiently explored in image tampering localization. Most existing methods employ a fixed decoder, where all images follow the same feature fusion and prediction pathway. This design is simple and stable but may be suboptimal for multi-type manipulation. Splicing, inpainting, and copy-move forgeries rely on different forensic evidence. A decoder that is sensitive to boundary discontinuity may perform well on splicing but may not be sufficient for copy-move, where the decisive cue is source–target homology. Conversely, a similarity-oriented branch may activate incorrectly on repetitive authentic textures if it is not guided by tampering type and pseudo-label reliability.

In summary, traditional forensic methods provide interpretable statistical cues but have limited robustness under complex post-processing. Deep learning based methods improve pixel-level localization but often rely on fixed binary segmentation pipelines. Multi-scale, multi-view, frequency-domain, and boundary-aware methods further enhance feature representation, yet they do not fully address source–target modeling for copy-move forgery or type-aware decoding for heterogeneous manipulation traces. Existing copy-move methods can capture duplicated-region similarity, but they are often separated from generic manipulation localization and may require stronger source-region supervision. These observations motivate the proposed pseudo-label-driven and dynamic multi-branch decoding framework. By generating weak source–target pseudo-labels, assigning confidence-aware supervision, dynamically fusing type-specific decoding branches, and refining small-region boundaries, the proposed method aims to provide a more adaptive and robust solution for image tampering region localization.

3. Pseudo-Label-Driven Image Manipulation Localization with Dynamic Multi-Branch Decoding

3.1 Overall Framework of the Proposed Method

Image manipulation localization aims to distinguish tampered pixels from pristine pixels at the pixel level [26]. However, different manipulation types usually leave heterogeneous forensic traces. Splicing and inpainting operations tend to introduce boundary discontinuities, texture inconsistencies, or local noise discrepancies, whereas copy-move manipulation is more difficult to identify because the copied target region and its source region originate from the same image. As a result, these two regions often share similar color distribution, noise pattern, and imaging conditions. If all manipulation types are forced to follow a single regression pathway, the model may learn mixed and ambiguous feature representations, especially when copy-move samples and other manipulation types are jointly involved in training. Therefore, image manipulation localization should not be formulated only as a conventional binary segmentation task. It is also necessary to model the source–target correspondence in copy-move manipulation and to adapt the decoding process to different manipulation characteristics.

Given an input RGB image $I \in \mathrm{R}^{H\times W\times 3}$, where H and W denote the height and width of the image, the proposed framework produces three outputs: A tampered-region probability map $\widehat{M}^t \in [0,1]^{H\times W}$, a copy-move source–target similarity map $\widehat{M}^s \in [0,1]^{H\times W}$, and a manipulation-type response vector $\hat{\tau}$. The overall mapping can be formulated as

$$\left(\widehat{M}^t, \widehat{M}^s, \hat{\tau}\right) = \Phi_\Theta(I) \quad (1)$$

Where $\Phi_\Theta(\cdot)$ denotes the whole network parameterized by Θ. Instead of directly mapping the input image to a single binary mask, Eq. (1) defines a multi-output localization framework in which general tampered-region prediction, copy-move source–target modeling, and manipulation-type-aware decoding are jointly considered.

To preserve both low-level forensic traces and high-level semantic information, the input image is first processed by a multi-scale encoder [27]. Low-level features are useful for capturing local artifacts such as weak boundaries and texture changes, while high-level features provide contextual information for distinguishing semantically plausible tampered regions from authentic image content. The encoded feature at the i-th scale is defined as

$$F_i = E_i(I;\theta_e), \quad i=1,2,\ldots,L \quad (2)$$

Where $E_i(\cdot)$ denotes the feature extraction function at the i-th scale, θ_e represents the encoder parameters, and L is the number of feature scales. Since small tampered regions may be weakened or even suppressed in deep semantic layers, the multi-scale representation in Eq. (2) provides a necessary basis for subsequent feature enhancement and decoding.

For copy-move manipulation, the ground-truth tampered mask usually indicates only the modified target region, while the corresponding source region is not explicitly annotated [28]. This lack of source–target supervision limits the model's ability to learn homologous regional relationships. To address this issue, a pseudo-label generation module is introduced during training. It uses the input image, the tampered-region prior, and the extracted multi-scale features to estimate the source–target pseudo-label for copy-move samples:

$$(\widetilde{M}^s, q^s) = P_\phi\left(I, \bar{M}^t, \{F_i\}_{i=1}^L\right) \quad (3)$$

Where $P_\phi(\cdot)$ denotes the pseudo-label generation function, $\widetilde{M}^s$ is the generated source–target pseudo-label, and $q^s \in [0,1]$ denotes its confidence score. The term $\bar{M}^t$ represents the

tampered-region prior, which can be provided by the ground-truth mask during supervised training or by the current prediction during iterative updating. By introducing $\widetilde{M}^{s}$, the model is guided to learn not only where the tampered region is located, but also which pristine region may have a homologous relationship with the copied target region.

After multi-scale encoding, the extracted features are further processed by a feature enhancement module. The purpose of this module is to strengthen manipulation-related cues by integrating local forensic traces, structural dependencies, and semantic context. The enhanced feature at the i-th scale is written as

$$G_i=A_i(F_i,G_{i-1};\theta_a), \quad i=1,2,\ldots,L \tag{4}$$

Where G_i denotes the enhanced feature, $A_i(\cdot)$ represents the feature enhancement function, and θ_a is its parameter set. When i=1, G_{i-1} can be initialized by the shallow feature or omitted. Through Eq. (4), fine-grained boundary information and high-level semantic responses are progressively integrated, making the learned representation more suitable for manipulation localization.

The complete architecture follows a unified pipeline consisting of multi-scale encoding, pseudo-label generation, feature enhancement, dynamic multi-branch decoding, and joint optimization. Figure 1 illustrates the overall network structure of the proposed method. The input image is first encoded into hierarchical features. These features are used in two directions: One path estimates copy-move source–target pseudo-labels for weak supervision, while the other path enhances manipulation-sensitive representations for localization. The enhanced features are then fed into a dynamic multi-branch decoder, which adaptively assigns different decoding branches to different manipulation characteristics. Finally, the network outputs the tampered-region probability map and the copy-move source–target similarity map.

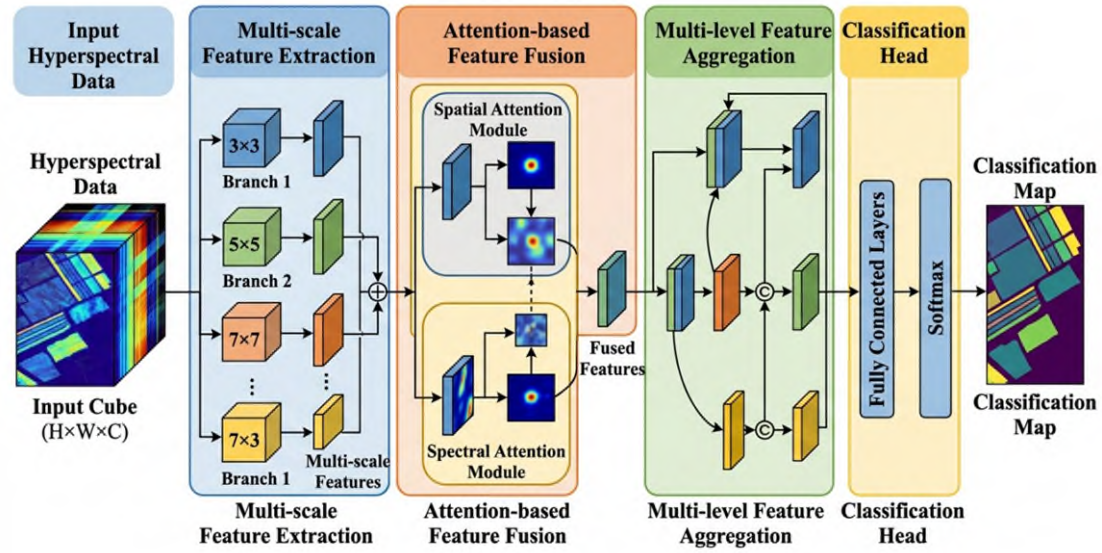


Figure 1 Overall network architecture of the proposed method

Unlike a fixed single-branch decoder, the dynamic multi-branch decoder explicitly separates different localization cues into multiple decoding branches. Let $K=\{1,2,\ldots,K\}$ denote the set of dynamic branches. The output response of the k-th decoding branch is formulated as

$$Z_k=D_k\left(\{G_i\}_{i=1}^{L};\theta_k\right),\quad k\in K \tag{5}$$

Where $D_k(\cdot)$ denotes the k-th decoding branch, θ_k represents its parameters, and Z_k is the corresponding response map. Different branches are expected to focus on different manipulation cues, such as source–target similarity, boundary inconsistency, local texture disturbance, or generic tampering evidence. To prevent branch responses from being fused in a fixed and non-adaptive manner, a manipulation-type-aware gating function is used to generate dynamic branch weights:

$$\alpha_k=\frac{exp\left(g_k\right)}{\sum_{j=1}^{K}exp\left(g_j\right)},\quad g=G\left(G_L;\theta_g\right) \tag{6}$$

Where α_k is the fusion weight of the k-th branch, g_k is the corresponding gating response, $G(\cdot)$ denotes the gating function, and θ_g represents the gating parameters. In this way, the model can adaptively emphasize the branch that is more consistent with the current manipulation pattern. For example, when the input image contains copy-move manipulation, the branch related to source–target correspondence is expected to receive a larger weight.

When the image contains splicing or inpainting, branches sensitive to boundary artifacts or local inconsistency should contribute more to the final prediction.

The final tampered-region prediction is obtained by combining the generic tampering response and the dynamically weighted branch responses. Meanwhile, the source–target similarity map is predicted by a dedicated copy-move decoding branch:

$$\widehat{M}^{t}=\sigma\left(Z_0+\sum_{k=1}^{K}\alpha_k Z_k\right),\quad \widehat{M}^{s}=\sigma\left(D_s\left(\{G_i\}_{i=1}^{L};\theta_s\right)\right) \tag{7}$$

Where Z_0 is the output of the generic tampered-region decoding branch, $D_s(\cdot)$ denotes the copy-move source–target decoding branch, θ_s is its parameter set, and $\sigma(\cdot)$ is the Sigmoid activation function. The two outputs are complementary. The map $\widehat{M}^{t}$ provides the final localization result for the tampered region, whereas $\widehat{M}^{s}$ supplies additional source–target correspondence constraints for copy-move manipulation. This design allows the source-region information to assist the recovery of target-region boundaries and region completeness.

To optimize all components in a unified manner, the training objective combines tampered-region localization, source–target pseudo-label supervision, boundary refinement, and dynamic branch regularization. The overall loss function is defined as

$$\begin{aligned}\mathrm{L}\quad=&\lambda_t \mathrm{L}_t\left(\widehat{M}^{t},M^{t}\right)+\lambda_s q^{s}\mathrm{L}_s\left(\widehat{M}^{s},\widetilde{M}^{s}\right)\\&+\lambda_b \mathrm{L}_b\left(\widehat{M}^{t},M^{t}\right)+\lambda_d \mathrm{L}_d(\hat{\tau},\tau)\end{aligned} \tag{8}$$

Where M^{t} denotes the ground-truth tampered-region mask, L_t is the tampered-region localization loss, L_s is the source–target pseudo-label supervision loss, L_b is the boundary refinement loss, and L_d is the dynamic branch regularization loss. The variable $\hat{\tau}$ denotes the manipulation-type label or type prior available during training. The coefficients λ_t, λ_s, λ_b, and λ_d balance different loss terms. The confidence score q^{s} is explicitly introduced to reduce the negative influence of unreliable pseudo-labels. Therefore, the proposed method does not optimize isolated manipulation cues separately, but establishes a unified learning objective across pixel-level localization, source–target relationship modeling, boundary refinement, and manipulation-type-aware decoding.

For clarity, the main symbols used in this section are summarized in Table 1.

Table 1 Definitions of Main Symbols and Variables

Symbol	Definition
$I \in \mathrm{R}^{H \times W \times 3}$	Input RGB image
M^t	Ground-truth tampered-region mask
$\widehat{M}^t$	Predicted tampered-region probability map
$\widehat{M}^s$	Predicted copy-move source–target similarity map
$\hat{\tau}$	Predicted manipulation-type response vector
F_i	Encoded feature at the i-th scale
G_i	Enhanced feature at the i-th scale
$\bar{M}^s$	Pseudo-label of the copy-move source–target regions
q^s	Confidence score of the pseudo-label
$D_k(\cdot)$	The k-th dynamic decoding branch
α_k	Dynamic fusion weight of the k-th decoding branch
$D_s(\cdot)$	Copy-move source–target decoding branch
L	Overall training loss

3.2 Pseudo-Label-Driven Copy-Move Source–Target Region Learning

Copy-move manipulation differs from splicing and inpainting in that the pasted target region is copied from another region within the same image. This property reduces the inconsistency of illumination, color distribution, and imaging noise between the tampered region and its surrounding content, making copy-move localization difficult when only generic tampering traces are considered. More importantly, most image manipulation localization datasets provide only the target tampered mask, while the corresponding copied source region is usually not annotated. Under this supervision condition, the model can learn where the pasted target region is located, but it lacks explicit guidance for discovering where the copied content originates. Therefore, copy-move localization requires an additional learning mechanism that can transform the implicit source–target homology into usable supervision signals.

To address this limitation, a pseudo-label-driven source–target learning strategy is introduced for copy-move samples. The key idea is to first use the tampered-region prior to identify the possible target region, then search for highly correlated regions within the same image, and finally generate a source–target pseudo-label with a confidence score. In this way,

source-region information can be incorporated into training even when manually annotated source masks are unavailable. Let $\bar{M}^t$ denote the tampered-region prior used in the current training stage. It can be derived from the ground-truth tampered mask at the early supervised stage or from the model prediction during iterative updating. Based on the enhanced feature map G, the feature representation of the candidate target region is obtained by mask-guided aggregation:

$$v_t=\frac{\sum_{x\in\Omega}\bar{M}^t(x)G(x)}{\sum_{x\in\Omega}\bar{M}^t(x)+\varepsilon} \quad (9)$$

Where x denotes a pixel position, Ω is the image lattice, $G(x)$ represents the enhanced feature at position x, and ε is a small constant for numerical stability. Eq. (9) extracts the feature prototype of the suspected pasted region, which serves as the query representation for searching homologous source candidates.

Since the source and target regions in copy-move manipulation are expected to share similar visual and feature distributions, the source candidate can be estimated by calculating the feature similarity between the target prototype and each spatial position in the same image. Before similarity calculation, the feature vector is normalized to reduce the influence of feature magnitude. The pixel-wise source likelihood map is formulated as

$$S(x)=\frac{\tilde{G}(x)^{\top}\tilde{v}_t}{\|\tilde{G}(x)\|_2\|\tilde{v}_t\|_2} \quad (10)$$

Where $\tilde{G}(x)$ and $\tilde{v}_t$ denote the normalized feature vector and normalized target prototype, respectively. A higher value of $S(x)$ indicates that the corresponding pixel is more likely to share homologous information with the candidate target region. However, directly selecting the most similar pixels may cause the target region itself to be repeatedly selected. Therefore, the source candidate should be searched outside the dilated target prior. The source pseudo-mask is generated as

$$\tilde{M}^{src}(x)=I(S(x)>\delta_s)\cdot\left(1-D\left(\bar{M}^t\right)(x)\right) \quad (11)$$

Where $I(\cdot)$ is the indicator function, δ_s is the similarity threshold, and $D(\cdot)$ denotes morphological dilation. The dilation operation excludes pixels near the suspected target region, reducing the risk of treating the pasted area itself as its source.

The reliability of the generated source candidate varies across samples. For copy-move images with clear duplicated structures, the similarity map usually contains a compact and high-response source region. For samples with severe post-processing, small pasted regions, or repetitive background textures, false source candidates may be produced. To suppress noisy pseudo-labels, a confidence score is assigned to each generated source–target pseudo-label [29]:

$$q^s=\frac{1}{|\Omega_s|}\sum_{x\in\Omega_s} S(x)\cdot exp\left(-\frac{|\Omega_s-\Omega_t|}{|\Omega_t|+\varepsilon}\right) \quad (12)$$

Where $\Omega_s=\{x|\widetilde{M}^{src}(x)=1\}$ denotes the candidate source region, and $\Omega_t=\{x|\bar{M}^t(x)=1\}$ denotes the target prior region. The first term evaluates the average feature similarity of the selected source region, while the exponential term penalizes abnormal area discrepancy between the source and target regions. Thus, q^s simultaneously reflects feature-level consistency and region-level plausibility.

After the source candidate and confidence score are obtained, the source–target pseudo-label is constructed by combining the candidate source region and the target prior:

$$\widetilde{M}^s(x)=I\left(\widetilde{M}^{src}(x)+\bar{M}^t(x)>0\right) \quad (13)$$

The resulting $\widetilde{M}^s$ provides weak supervision for the source–target decoding branch. Unlike the ordinary tampered-region mask, $\widetilde{M}^s$ does not merely indicate the pasted target region; it describes the union of the copied source region and the pasted target region, allowing the model to learn copy-move homology explicitly [30].

The generation and refinement process of $\widetilde{M}^s$ is not limited to a single pass. As the localization branch becomes more stable, the predicted tampered-region probability map can provide a more adaptive target prior for pseudo-label updating. Therefore, an iterative update strategy is adopted to gradually refine the target prior:

$$\bar{M}^t_{r+1}=I\left(\eta M^t+(1-\eta)\widehat{M}^t_r>\delta_t\right) \quad (14)$$

Where r denotes the update round, $\widehat{M}^t_r$ is the predicted tampered-region probability map in the r-th round, $\eta\in[0,1]$ controls the contribution of the ground-truth mask and the current prediction, and δ_t is the binarization threshold. At the early stage, a larger η can stabilize training with reliable target supervision. At later stages, a smaller η allows the model

prediction to participate more in pseudo-label generation, which helps refine source candidates when the initial target prior is incomplete or coarse.

The full procedure is summarized in Figure 2. The input image is first encoded to obtain manipulation-sensitive features, and the target prior is then used to extract the candidate target prototype. The similarity search module estimates source candidates outside the target region, while the confidence evaluation module filters unreliable pseudo-labels. During iterative training, the predicted tampered mask updates the target prior, and the refined source–target pseudo-label is used to supervise the source–target decoding branch.

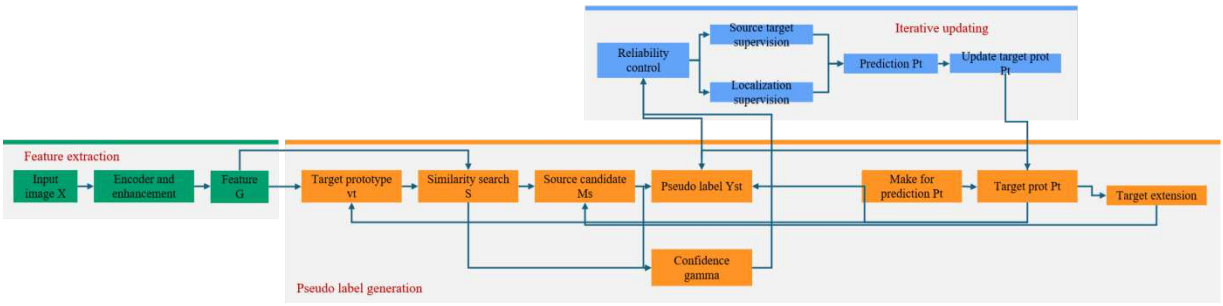


Figure 2 Copy-move source–target pseudo-label generation and iterative updating process

To further strengthen the homology constraint between the source and target regions, feature-level consistency is imposed on the generated pseudo-labels. The target and source feature vectors are extracted as

$$V_{tar}=\frac{\sum_{x\in\Omega}\dot{M}^{t}(x)G(x)}{\sum_{x\in\Omega}\dot{M}^{t}(x)+\varepsilon},\quad V_{src}=\frac{\sum_{x\in\Omega}\dot{M}^{src}(x)G(x)}{\sum_{x\in\Omega}\dot{M}^{src}(x)+\varepsilon} \tag{15}$$

Because copy-move source and target regions originate from the same image content, their feature distance should be reduced during training [31]. The source–target consistency loss is therefore defined as

$$L_{st}=-q^{s}\log\left(\frac{V_{tar}^{\top}V_{src}}{\|V_{tar}\|_2\|V_{src}\|_2}+\varepsilon\right) \tag{16}$$

The confidence score q^{s} is introduced into Eq. (16) to suppress the contribution of unreliable pseudo-labels. When the generated source candidate is reliable, the loss encourages the model to pull the source and target features closer in the embedding space. When the

source candidate is uncertain, the loss contribution is weakened, reducing the risk of error accumulation during iterative training.

Different training samples provide different levels of supervision for source–target learning. Therefore, the source–target branch is not forced to follow a single training strategy for all samples. Instead, the supervision signal is determined according to the manipulation type and pseudo-label reliability, as summarized in Table 2.

Table 2 Supervision Settings for Different Types of Training Samples

Training sample type	Source–target supervision signal	Loss setting	Training strategy
Copy-move sample with reliable pseudo-label	$\bar{M}^s$	$\mathcal{L}_s + \mathcal{L}_{st}$	Use the generated pseudo-label to supervise the source–target branch and impose source–target consistency.
Copy-move sample with low-confidence pseudo-label	$\bar{M}^s$ weighted by q^s	$q^s\mathcal{L}_s + q^s\mathcal{L}_{st}$	Reduce the influence of unreliable pseudo-labels during optimization.
Copy-move sample without valid source candidate	Not used temporarily	Disabled	Use only tampered-region localization supervision and update pseudo-labels in later iterations.
Non-copy-move tampered sample	0	Negative supervision for source–target branch	Suppress false source–target responses for non-copy-move manipulations.
Pristine sample	0	Negative supervision	Prevent the source–target branch from activating on authentic images.

Through this strategy, copy-move source–target learning is incorporated without requiring complete manual source-region annotations. The pseudo-label generation module provides weak supervision for homologous region discovery, the confidence mechanism reduces the influence of noisy source candidates, and the iterative update process gradually improves pseudo-label quality. As a result, the source–target branch can assist the final tampered-region localization by supplying copy-move-specific structural cues rather than relying only on generic tampering artifacts.

3.3 Tamper-Type-Aware Dynamic Multi-Branch Decoder

Although the pseudo-label-driven source–target learning strategy provides additional supervision for copy-move manipulation, the final localization performance still depends on whether the decoder can transform the enhanced features into type-specific localization responses. Different manipulation types usually rely on different forensic cues. Copy-move manipulation requires the decoder to preserve source–target correspondence and regional homology, splicing manipulation is more related to boundary discontinuity and cross-region inconsistency, and inpainting manipulation often leaves local texture distortion or structural compensation traces. If these heterogeneous cues are decoded through a fixed and unified pathway, the decoder may produce an averaged representation that is insufficiently sensitive to manipulation-specific evidence. Therefore, the decoding stage should be able to adjust its feature selection and fusion strategy according to the manipulation characteristics of the current input.

To achieve this goal, a tamper-type-aware dynamic multi-branch decoder is constructed. Instead of assigning all samples to the same decoding pathway, the enhanced multi-scale features are first used to estimate type-aware responses, and then multiple decoding branches are dynamically weighted according to these responses. Let G_L denote the highest-level enhanced feature obtained from the feature enhancement module. A compact global descriptor is extracted by global average pooling:

$$\mathrm{h}=GAP(G_L) \tag{17}$$

Where h summarizes the global manipulation-related context of the input image. Since the global descriptor contains high-level semantic and forensic responses, it is used to estimate the probability-like type response over different manipulation patterns:

$$p=Softmax(W_\tau \mathrm{h}+b_\tau) \tag{18}$$

Where $p=[p_1,p_2,\ldots,p_C]$ denotes the tamper-type response vector, C is the number of considered manipulation types, and W_τ and b_τ are learnable parameters. The response p_C reflects the relative tendency of the input image toward the c-th manipulation type. This response is not used merely as an image-level classification result; Instead, it guides how different decoding branches participate in pixel-level localization.

The dynamic branch weights are generated by combining the type response with the reliability of the copy-move source–target pseudo-label [32]. This design is necessary because a copy-move branch should contribute more strongly when the input is likely to contain copy-move manipulation and the generated source–target pseudo-label is reliable. The branch-weight vector is calculated as

$$\alpha=Softmax\left(\frac{W_g[p;q^s]+b_g}{T}\right) \quad (19)$$

Where $\alpha=[\alpha_1,\alpha_2,\ldots,\alpha_K]$, K is the number of decoding branches, $[p;q^s]$ denotes the concatenation of the type response vector and the pseudo-label confidence score, W_g and b_g are learnable parameters of the gating module, and T is the temperature coefficient. A smaller T makes the branch selection sharper, while a larger T encourages smoother fusion among branches. By introducing q^s, unreliable source–target pseudo-labels are prevented from excessively activating the copy-move-related branch.

Each decoding branch receives the enhanced multi-scale features and produces a branch-specific localization response. To avoid repeating the same decoding function across branches, each branch is assigned a distinct transformation function that emphasizes different manipulation cues:

$$R_k=D_k\left(\{G_i\}_{i=1}^{L};\theta_k\right), \quad k=1,2,\ldots,K \quad (20)$$

Where R_k denotes the response map of the k-th decoding branch, $D_k(\cdot)$ is the corresponding branch decoder, and θ_k denotes its learnable parameters. In practice, one branch can be biased toward source–target similarity, another toward boundary inconsistency, and another toward generic tampering traces. The decoder does not require a hard assignment of samples to a single branch. Instead, branch responses are fused according to the dynamic weights:

$$R_{dyn}=\sum_{k=1}^{K}\alpha_k R_k \quad (21)$$

Through Eq. (21), the decoder can adaptively emphasize the response that best matches the current manipulation characteristics. For copy-move samples, the source–target branch tends to provide stronger structural constraints. For splicing or inpainting samples, branches focusing on boundary inconsistency and local texture disturbance can contribute more to the final prediction.

The structure of the tamper-type-aware dynamic decoder is organized around this weighted decoding mechanism. After multi-scale enhanced features are fed into several parallel decoding branches, the type-aware gating module estimates branch weights from global manipulation responses and pseudo-label confidence. The weighted responses are then fused with the generic tampering response to produce the final tampered-region map. Figure 3 presents this decoding process, where the type response, branch weight generation, parallel decoding, and dynamic feature fusion are connected within a unified decoder.

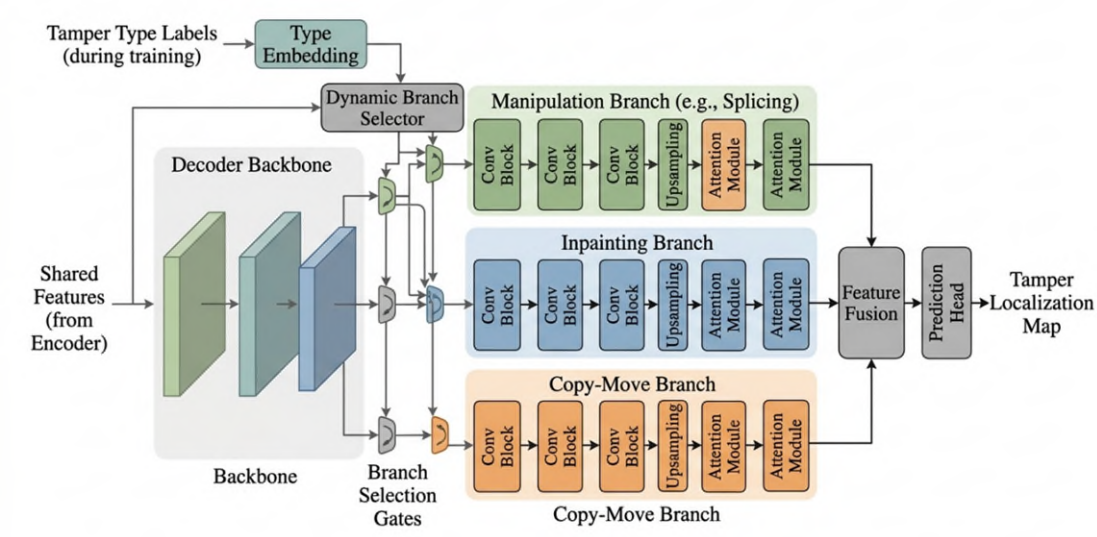


Figure 3 Structure of the tamper-type-aware dynamic multi-branch decoder

To retain the robustness of generic tampering localization, the dynamic response is not used alone. A generic decoding branch is preserved to capture manipulation-independent localization cues, and the final tampered-region prediction is obtained by fusing the generic response and the dynamic response:

$$\widetilde{M}^{t}=\sigma\left(R_{0}+\gamma R_{dyn}\right) \tag{22}$$

Where R_0 denotes the response of the generic tampered-region decoder, γ is a learnable or predefined balancing coefficient, and $\sigma(\cdot)$ denotes the Sigmoid activation function. The generic branch stabilizes localization for manipulation types with weak type evidence, while the dynamic branch improves sensitivity to type-specific forensic cues [33].

Since copy-move localization further requires source–target relationship modeling, the response of the source–target branch is also used to guide feature interaction between $\widetilde{M}^{t}$ and the source–target map. A type-aware correction term is introduced as

$$\widehat{M}_{ref}^{t}=\sigma\left(Conv\left([\widehat{M}^{t},\widehat{M}^{s},\alpha_s R_s]\right)\right) \quad (23)$$

Where $\widehat{M}_{ref}^{t}$ denotes the refined tampered-region probability map, R_s represents the source–target branch response, α_s is the dynamic weight assigned to the source–target branch, and [·] denotes channel-wise concatenation. This refinement allows source–target evidence to contribute to the final localization only when the gating module assigns sufficient confidence to the copy-move-related branch.

To prevent the dynamic decoder from collapsing into a single dominant branch for all samples, a branch diversity regularization term is imposed on the branch responses:

$$\mathrm{L}_{div}=\frac{1}{K(K\text{-}1)}\sum_{i=1}^{K}\sum^{K}\frac{\langle R_i,R_j\rangle}{\|R_i\|_2\|R_j\|_2+\epsilon} \quad (24)$$

This term penalizes excessive similarity among different branch responses, encouraging each branch to preserve manipulation-specific decoding behavior. Meanwhile, to ensure that branch selection is consistent with the type-aware response, a gating supervision term is introduced:

$$\mathrm{L}_{gate}=-\sum_{c=1}^{C} y_c \log\left(p_c+\epsilon\right) \quad (25)$$

Where y_c is the type label or type prior of the input image. For non-copy-move samples without reliable type labels, y_c can be replaced by a weak type prior derived from dataset construction or augmentation metadata. The two regularization terms are combined with the localization objective in later optimization, enabling the decoder to maintain both type sensitivity and localization stability.

The dynamic branch selection and feature fusion process is summarized in Algorithm 1. Given the enhanced multi-scale features, the algorithm first extracts a global descriptor and estimates the tamper-type response. Then, the branch weights are calculated by the gating module. Each decoding branch generates its own response map, and the weighted response is fused with the generic tampering response. Finally, the source–target response is used to refine the tampered-region prediction when the copy-move branch receives sufficient activation.

Algorithm 1 Dynamic Branch Selection and Feature Fusion Algorithm

Algorithm 1 Dynamic Branch Selection and Feature Fusion Algorithm

Require: Enhanced multi-scale features $\{G_i\}_{i=1}^{L}$, pseudo-label confidence γ, branch decoders $\{D_k\}_{k=1}^{K}$, generic decoder D_g

Ensure: Refined tampered-region probability map $\hat{M}_{ref}$ and branch weights α

Select the highest-level enhanced feature G_L.
Obtain the global descriptor by average pooling: $h \leftarrow \mathrm{GAP}(G_L)$.
Estimate the tamper-type response: $q \leftarrow \mathrm{Softmax}(W_t h + b_t)$.
Concatenate the type response and pseudo-label confidence: $z \leftarrow [q, \gamma]$.
Generate the gating logits: $s \leftarrow W_g z + b_g$.
Compute dynamic branch weights: $\alpha \leftarrow \mathrm{Softmax}(s/T)$.
Initialize an empty branch-response set $\mathscr{R}$.
for $k = 1$ to K **do**
    Feed enhanced multi-scale features into the k-th branch.
    Compute branch-specific response: $R_k \leftarrow D_k(\{G_i\}_{i=1}^{L})$.
    Store R_k in $\mathscr{R}$.
end for
Initialize the dynamic response: $R_{dyn} \leftarrow 0$.
for $k = 1$ to K **do**
    Fuse the weighted branch response: $R_{dyn} \leftarrow R_{dyn} + \alpha_k R_k$.
end for
Compute the generic tampering response: $R_g \leftarrow D_g(\{G_i\}_{i=1}^{L})$.
Fuse generic and dynamic responses: $R_f \leftarrow R_g + \lambda R_{dyn}$.
Obtain the preliminary localization map: $\hat{M} \leftarrow \sigma(R_f)$.
Extract the copy-move branch response R_{cm} from $\mathscr{R}$.
Read the copy-move branch weight α_{cm} from α.
if $\alpha_{cm} > \tau_{cm}$ **then**
    Refine localization using source–target evidence: $\hat{M}_{ref} \leftarrow \sigma(\mathrm{Conv}([\hat{M}, \alpha_{cm} R_{cm}]))$.
else
    Keep the preliminary prediction: $\hat{M}_{ref} \leftarrow \hat{M}$.
end if
return $\hat{M}_{ref}$ and α.

Through the above design, the decoder no longer treats all manipulation types as identical pixel-level regression targets. Instead, the global type response determines the relative contribution of each decoding branch, while the source–target confidence controls the participation of copy-move-specific evidence. This mechanism connects the pseudo-label-driven source–target learning in Section 3.2 with the final tampered-region localization, allowing the model to jointly exploit generic tampering artifacts and manipulation-specific decoding cues.

3.4 Class-Imbalance Optimization and Boundary Refinement Loss for Small Tampered Regions

Small tampered regions are particularly difficult to localize because they occupy only a limited portion of the image while still being able to alter the semantic integrity of the visual content. In pixel-level supervision, most pixels usually belong to the pristine category, whereas tampered pixels may account for only a small fraction of the training image. If all

pixels are treated with equal importance, the optimization process tends to be dominated by pristine pixels, and the model may obtain a low training loss by predicting most pixels as authentic. This imbalance is more harmful for small-area manipulation, where weak tampering traces and thin boundaries can be easily suppressed during downsampling and feature aggregation. Therefore, the loss function should not only supervise pixel-wise classification, but also increase the contribution of minority tampered pixels and explicitly preserve boundary consistency.

Let Ω denote the image lattice and $x \in \Omega$ denote a pixel position. For binary tampered-region localization, the ground-truth label at pixel x can be represented by a one-hot vector:

$$y_x^c = I(M^t(x) = c), \quad c \in 0,1 \tag{26}$$

Where c=0 indicates the pristine category and c=1 indicates the tampered category. The refined prediction $\widehat{M}_{ref}^t$ generated by the dynamic decoder is used as the final tampered-region probability map, and its two-class probability representation is denoted as p_x^c. Since the pristine category often has much more pixels than the tampered category, the class proportion is first estimated from the ground-truth mask:

$$\rho_c = \frac{\sum_{x \in \Omega} y_x^c + \varepsilon}{|\Omega| + 2\varepsilon}, \quad c \in 0,1 \tag{27}$$

Where $|\Omega|$ denotes the total number of pixels and ε is a small constant used for numerical stability. A smaller ρ_c indicates that the corresponding category is underrepresented in the current image. To prevent the minority class from being overwhelmed by the majority class, a category-balanced weight is assigned to each class [34]:

$$\omega_c = \frac{1/(\rho_c + \varepsilon)}{\sum_{j=0}^{1} 1/(\rho_j + \varepsilon)} \tag{28}$$

Through Eq. (28), the tampered class receives a larger weight when its area ratio is small. This weighting mechanism allows the model to focus more on small manipulated regions instead of being optimized mainly by the large number of pristine pixels. Based on the category-balanced weights, the pixel-level localization loss is defined as

$$L_{cb} = -\frac{1}{|\Omega|} \sum_{x \square \Omega} \sum_{c=0}^{1} \omega_c y_x^c \log(p_x^c + \varepsilon) \tag{29}$$

Although L_{cb} improves the contribution of small tampered regions, pixel-wise classification alone may still produce incomplete masks or blurred boundaries. This is because the loss treats each pixel independently and does not directly constrain the overlap structure of the predicted region. To improve region-level consistency, a soft Dice loss is introduced:

$$L_{dice}=1-\frac{2\sum_{x\in\Omega}\widehat{M}^{t}_{rel}(x)M^{t}(x)+\varepsilon}{\sum_{x\in\Omega}\widehat{M}^{t}_{rel}(x)+\sum_{x\in\Omega}M^{t}(x)+\varepsilon} \quad (30)$$

The Dice term emphasizes the overlap between the predicted tampered region and the ground-truth mask, which is beneficial for small-object localization because it directly penalizes missing tampered pixels. However, accurate localization of manipulated regions also depends on boundary precision [35]. In image forensics, the boundary between tampered and pristine regions often contains critical traces such as discontinuous textures, unnatural transitions, and local inconsistencies. Therefore, a boundary map is constructed by using the morphological gradient of the tampered mask:

$$B(M)=D(M)-E(M) \quad (31)$$

Where $D(\cdot)$ and $E(\cdot)$ denote morphological dilation and erosion, respectively. The boundary map B(M) highlights the contour region of the tampered mask. By applying the same operation to both the prediction and the ground-truth mask, boundary-level supervision can be imposed as

$$L_{bd}=1-\frac{2\sum_{x\in\Omega}B(\widehat{M}^{t}_{ref})(x)B(M^{t})(x)+\varepsilon}{\sum_{x\in\Omega}B(\widehat{M}^{t}_{ref})(x)+\sum_{x\in\Omega}B(M^{t})(x)+\varepsilon} \quad (32)$$

The boundary refinement loss in Eq. (32) encourages the predicted contour to align with the real tampered boundary. Compared with a pure pixel-level loss, this term is more sensitive to edge deviation and can reduce boundary expansion, shrinkage, and fragmentation. It is particularly useful for small tampered regions, where a slight boundary shift may cause a large reduction in localization accuracy.

The final loss for small-region optimization and boundary refinement is obtained by combining category-balanced pixel supervision, region-overlap supervision, and boundary-level supervision:

$$L_{small}=\lambda_{cb}L_{cb}+\lambda_{dice}L_{dice}+\lambda_{bd}L_{bd} \quad (33)$$

Where λ_{cb}, λ_{dice}, and λ_{bd} are weighting coefficients. The category-balanced term improves the contribution of minority tampered pixels, the Dice term enhances region completeness, and the boundary term refines the contour structure. These three terms complement the pseudo-label-driven source–target learning and dynamic multi-branch decoding described in the previous sections by providing more stable supervision for small and weak tampered regions.

The computation process of the proposed joint loss is organized in Algorithm 2. Given the refined tampered-region prediction and the ground-truth mask, the algorithm first estimates class proportions and assigns category-balanced weights. It then calculates the weighted pixel-level loss, computes the Dice overlap loss, extracts boundary maps through morphological operations, and finally combines all loss terms into L_{small}.

Algorithm 2 Class-Balanced and Boundary-Refined Joint Loss Calculation

Algorithm 2 Class-Balanced and Boundary-Refined Joint Loss Calculation

Require: Refined prediction $\hat{M}_{ref}$, ground-truth mask M, image lattice Ω
Require: Loss weights λ_{cb}, λ_{dice}, λ_{bd}, and constant ε
Ensure: Joint small-region loss $\mathscr{L}_{small}$
Convert $\hat{M}_{ref}$ into two-class probabilities $p_x^1 = \hat{M}_{ref}(x)$ and $p_x^0 = 1 - \hat{M}_{ref}(x)$.
Initialize the category-balanced loss $\mathscr{L}_{cb} \leftarrow 0$.
for $c = 0$ to 1 **do**
    Construct the one-hot label y_x^c from the ground-truth mask M.
    Estimate the class proportion: $\rho_c \leftarrow \frac{1}{|\Omega|}\sum_{x\in\Omega} y_x^c$.
    Compute the class-balanced weight: $w_c \leftarrow \frac{1}{\log(1+\rho_c+\varepsilon)}$.
end for
for each pixel $x \in \Omega$ **do**
    for $c = 0$ to 1 **do**
        Accumulate the weighted pixel loss: $\mathscr{L}_{cb} \leftarrow \mathscr{L}_{cb} - w_c y_x^c \log(p_x^c + \varepsilon)$.
    end for
end for
Normalize $\mathscr{L}_{cb}$ by $|\Omega|$.
Compute the soft Dice loss: $\mathscr{L}_{dice} \leftarrow 1 - \frac{2\sum_{x\in\Omega}\hat{M}_{ref}(x)M(x)+\varepsilon}{\sum_{x\in\Omega}\hat{M}_{ref}(x)+\sum_{x\in\Omega}M(x)+\varepsilon}$.
Extract the ground-truth boundary: $B(M) \leftarrow \mathscr{D}(M) - \mathscr{E}(M)$.
Extract the predicted boundary: $B(\hat{M}_{ref}) \leftarrow \mathscr{D}(\hat{M}_{ref}) - \mathscr{E}(\hat{M}_{ref})$.
Compute the boundary refinement loss: $\mathscr{L}_{bd} \leftarrow \frac{1}{|\Omega|}\sum_{x\in\Omega}\left|B(\hat{M}_{ref})(x) - B(M)(x)\right|$.
Combine all loss terms: $\mathscr{L}_{small} \leftarrow \lambda_{cb}\mathscr{L}_{cb} + \lambda_{dice}\mathscr{L}_{dice} + \lambda_{bd}\mathscr{L}_{bd}$.
return $\mathscr{L}_{small}$.

By integrating class-balanced weighting, region-level overlap optimization, and boundary refinement, the loss function reduces the dominance of pristine pixels and improves

the sensitivity of the model to small tampered regions. As a result, the final prediction can better preserve both the completeness and the contour accuracy of manipulated regions.

4 Experimental Results and Analysis

4.1 Experimental Settings

To evaluate the proposed image manipulation localization method under both cross-dataset and intra-dataset conditions, two representative datasets were selected: COLUMBIA and COVER. The COLUMBIA dataset mainly contains splicing forgeries, where tampered regions are introduced from external images. This dataset is therefore suitable for evaluating whether the model can capture local inconsistencies caused by image composition. In contrast, the COVER dataset focuses on copy-move forgery, where the forged region and its source region come from the same image. This characteristic makes COVER more suitable for validating pseudo-label-driven source-target region learning. The dataset configuration and partition strategy are summarized in Table 3. The use of these two datasets follows the common practice of evaluating splicing and copy-move manipulation separately in image forensics research.

Table 3 Experimental datasets and data partition

Dataset	COLUMBIA	COVER
Main tampering type	Splicing	Copy-move
Forged images	180	100
Authentic images	183	100
Annotation type	Tampered-region mask / edge mask	Forged-region mask and duplicated-region mask
Training split	126 forged + 128 authentic	70 forged + 70 authentic
Validation split	27 forged + 27 authentic	15 forged + 15 authentic
Test split	27 forged + 28 authentic	15 forged + 15 authentic
Main use	Splicing localization and cross-dataset evaluation	Copy-move localization and cross-dataset evaluation

The experiments were divided into two main settings. In the cross-dataset setting, the model trained on one dataset was directly tested on the other dataset without additional fine-tuning, which was used to measure whether the learned representation could transfer

between splicing and copy-move scenarios. In the intra-dataset setting, each dataset was divided into training, validation, and testing subsets, which was used to evaluate the fitting ability of the model when training and testing data share the same distribution. This arrangement is particularly important because splicing and copy-move forgeries have different visual and forensic traces. Splicing usually introduces inconsistencies in illumination, boundary, texture, and noise, while copy-move forgery is more difficult because the copied and pasted regions may retain similar imaging characteristics.

The implementation settings are listed in Table 4. The input size was fixed at 512 × 512 pixels to balance localization accuracy and computational cost. A ConvNeXt-based encoder was used as the feature extraction backbone, and a dynamic multi-branch decoder was adopted to handle different tampering characteristics. The pseudo-label threshold was set to 0.70, and the boundary refinement loss weight was set to 0.20, which was later verified by the sensitivity analysis. The compared methods were selected according to the experimental design in the generated figures. MVSS-Net/MVSS-Net++ was used as a multi-view multi-scale supervised baseline, PAM-Net and DDBCA-PDF were used as attention-based and dual-decoding related baselines, and NCL and CSR-Net were included in intra-dataset comparison because they represent recent contrastive-learning and boundary-aware localization methods.

Table 4 Implementation details and model parameter settings

Item	Setting	Item	Setting
Input image size	512 × 512 pixels	Training epochs	80
Backbone	ConvNeXt-based encoder	Data augmentation	Random flipping, rotation, scaling, and contrast enhancement
Decoder structure	Dynamic multi-branch decoder	Evaluation threshold	0.5
Number of dynamic branches	3	Evaluation metrics	Precision, Recall, F1, AUC, IoU
Pseudo-label threshold	0.7	Cross-dataset comparison methods	MVSS-Net++, PAM-Net, DDBCA-PDF, Proposed
Boundary loss weight	0.2	Intra-dataset comparison methods	NCL, CSR-Net, DDBCA-PDF, Proposed

Optimizer	AdamW	Robustness comparison methods	MVSS-Net, PAM-Net, DDBCA-PDF, Proposed
Initial learning rate	1×10^{-4}	Complexity comparison methods	MVSS-Net, PAM-Net, DDBCA-PDF, Proposed, Proposed-Lite
Minimum learning rate	1×10^{-8}	Implementation language	Python
Batch size	4		

4.2 Cross-Dataset and Intra-Dataset Performance Comparison

The cross-dataset experiment was designed to test the generalization ability of different methods when the training and testing distributions are inconsistent. This is a challenging setting because the two selected datasets represent different tampering mechanisms. The results in Figure 4 show that the proposed method achieves the best overall performance on both COLUMBIA and COVER. On COLUMBIA, the proposed method obtains an F1 score of 75.4% and an AUC of 93.5%, exceeding DDBCA-PDF by 1.6 percentage points in F1 and 1.4 percentage points in AUC. Compared with MVSS-Net++, the improvement is more obvious, reaching 9.4 percentage points in F1 and 4.9 percentage points in AUC. This suggests that the proposed method can better identify splicing traces when transferred from a different dataset.

The COVER results in Figure 4 further demonstrate the value of source-target modeling for copy-move forgery localization. On this dataset, the proposed method reaches 52.7% F1 and 87.2% AUC. Although the absolute score is lower than that on COLUMBIA, this is reasonable because copy-move forgeries usually contain fewer external inconsistencies than splicing forgeries. The proposed method still improves over DDBCA-PDF by 3.6 percentage points in F1 and 2.4 percentage points in AUC. PAM-Net performs relatively well on COLUMBIA with 72.2% F1, but drops to 43.0% F1 on COVER, indicating that general attention enhancement alone may not be sufficient for copy-move localization. The proposed method maintains a clearer advantage on COVER, which supports the effectiveness of pseudo-label-driven source-target learning.

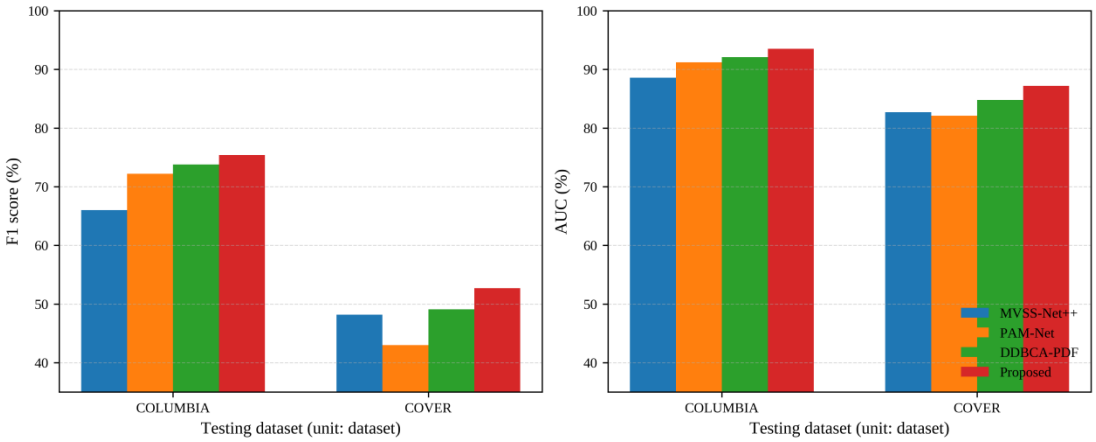


Figure 4 Comprehensive comparison of cross-dataset F1/AUC performance

The intra-dataset comparison provides another perspective by evaluating the model under matched training and testing distributions. As shown in Figure 5, all strong methods achieve higher scores than in the cross-dataset setting, but the proposed method still obtains the best performance. On COLUMBIA, the proposed method achieves 99.0% F1 and 99.9% AUC, slightly higher than DDBCA-PDF with 98.7% F1 and 99.9% AUC. On COVER, the proposed method obtains 98.6% F1 and 99.9% AUC, while DDBCA-PDF reaches 98.2% F1 and 99.9% AUC. Although the improvement over DDBCA-PDF is small in the intra-dataset setting, the proposed method remains consistently better on both splicing and copy-move tasks. More importantly, it greatly exceeds NCL and CSR-Net on COVER. Compared with NCL, the proposed method improves F1 by 13.6 percentage points, and compared with CSR-Net, the improvement reaches 20.6 percentage points.

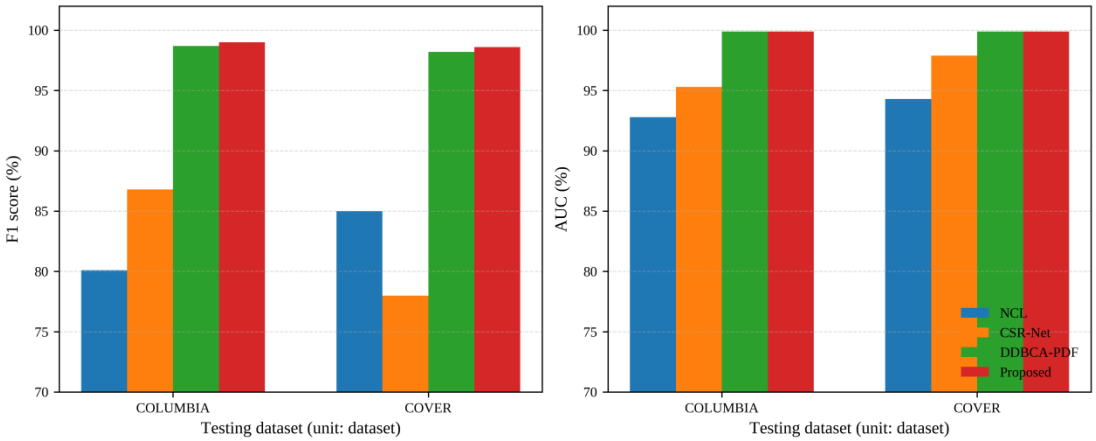


Figure 5 Comprehensive comparison of in-dataset F1/AUC performance

The tampering-type comparison in Figure 6 explains why the proposed method performs more robustly across the two datasets. For splicing on COLUMBIA, the proposed method

achieves 88.2% precision, 84.6% recall, 86.4% F1, and 76.1% IoU. Compared with DDBCA-PDF, the F1 score increases by 2.1 percentage points and the IoU increases by 3.3 percentage points. For copy-move on COVER, the proposed method achieves 85.0% precision, 82.8% recall, 83.9% F1, and 72.3% IoU, exceeding DDBCA-PDF by 3.4 percentage points in F1 and 4.9 percentage points in IoU. The improvement on copy-move is larger than that on splicing, which indicates that the source-target pseudo-label branch contributes more directly to copy-move localization. This result also shows that the proposed model does not only improve image-level discrimination, but also generates more accurate pixel-level masks for different manipulation types.

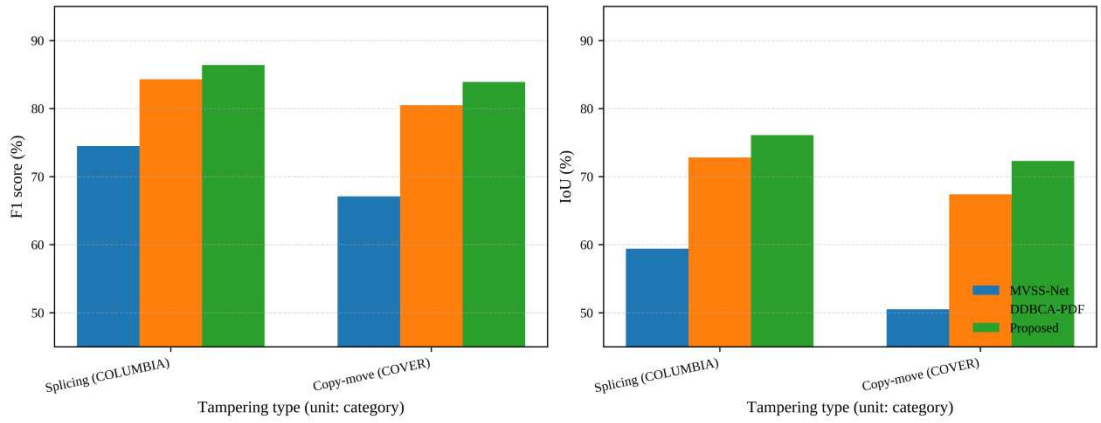


Figure 6 Localization performance comparison under different tampering types

4.3 Visual Analysis of Localization Results

The quantitative results in Section 5.2 indicate that the proposed method achieves stronger localization performance on both COLUMBIA and COVER, and the visual results in Figure 7 further explain this improvement from a qualitative perspective. For the COLUMBIA splicing sample, the tampered region is located on the right side of the image and has a relatively regular boundary. The baseline result can detect part of the manipulated area, but the predicted mask is incomplete and contains scattered false-positive regions. In contrast, the proposed method produces a much cleaner binary mask and its shape is closer to the ground truth. This observation is consistent with the cross-dataset and intra-dataset results, where the proposed method obtains 75.4% F1 and 93.5% AUC in the cross-dataset setting, and further reaches 99.0% F1 and 99.9% AUC in the intra-dataset setting on COLUMBIA.

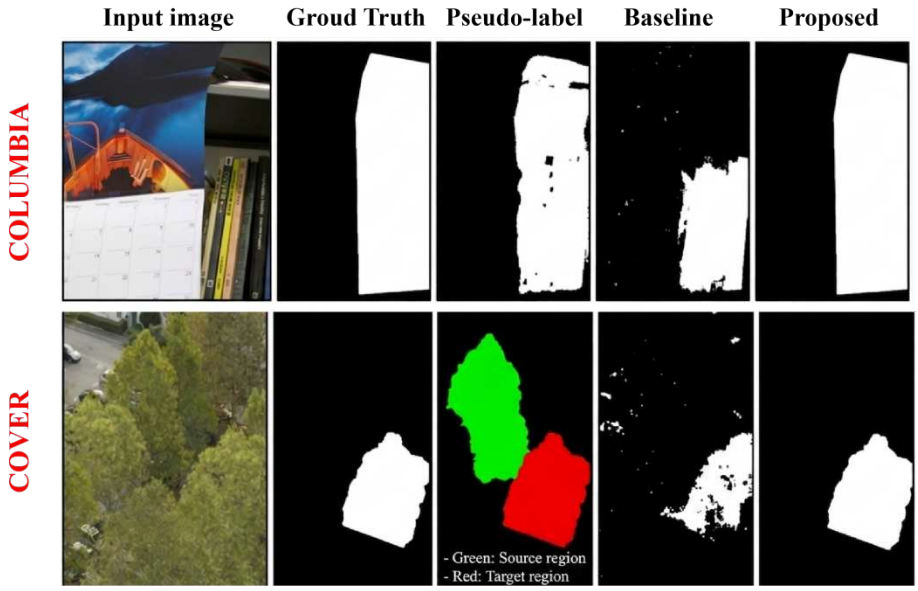


Figure 7 Visual comparison of localization results under different tampering types

For the COVER copy-move sample, the localization task is more difficult because the source and target regions come from the same image and share similar texture, color, and noise distributions. The pseudo-label visualization in Figure 7 uses green to represent the source region and red to represent the target region, which corresponds to the source-target modeling idea emphasized in the original PDF. Compared with the baseline, which produces obvious noise and inaccurate boundaries, the proposed method localizes the target region more compactly and suppresses many irrelevant responses. This visual difference supports the numerical results on COVER, where the proposed method improves the cross-dataset F1 score to 52.7% and achieves 83.9% F1 with 72.3% IoU in the copy-move type comparison. Overall, Figure 7 shows that the proposed framework improves not only metric-level performance, but also mask completeness, boundary consistency, and false-positive suppression.

4.4 Robustness, Ablation Study, and Complexity Analysis

Robustness analysis was conducted under common image degradation operations, including JPEG compression, image scaling, Gaussian blur, Gaussian noise, and social platform transmission. These operations are important because manipulated images are often compressed, resized, or re-uploaded before being detected in real applications. Under JPEG compression, the proposed method maintains the highest F1 score at all quality levels in

Figure 8. When the JPEG quality factor is reduced to 30, the proposed method still achieves 70.8% F1, while DDBCA-PDF obtains 68.6%, MVSS-Net obtains 61.5%, and PAM-Net obtains 60.2%. When the quality factor increases to 90, the proposed method reaches 81.0% F1. The curve is not strictly monotonic because compression artifacts may sometimes suppress local texture noise while still preserving boundary-level cues, but the proposed method remains consistently superior to the baselines.

The scaling results in Figure 8 show a similar trend. At the scaling factor of 0.50, the proposed method obtains 72.6% F1, which is higher than DDBCA-PDF by 3.4 percentage points and higher than MVSS-Net by 14.6 percentage points. At the original scale of 1.00, the proposed method reaches 82.1% F1. When the image is enlarged to 1.25 and 1.50, the F1 score decreases to 80.0% and 78.1%, respectively. This non-linear variation is reasonable because moderate scaling may preserve the main tampering region, while excessive interpolation can change local forensic traces. The performance gap between the proposed method and the baselines indicates that dynamic multi-branch decoding is more stable under resolution variation.

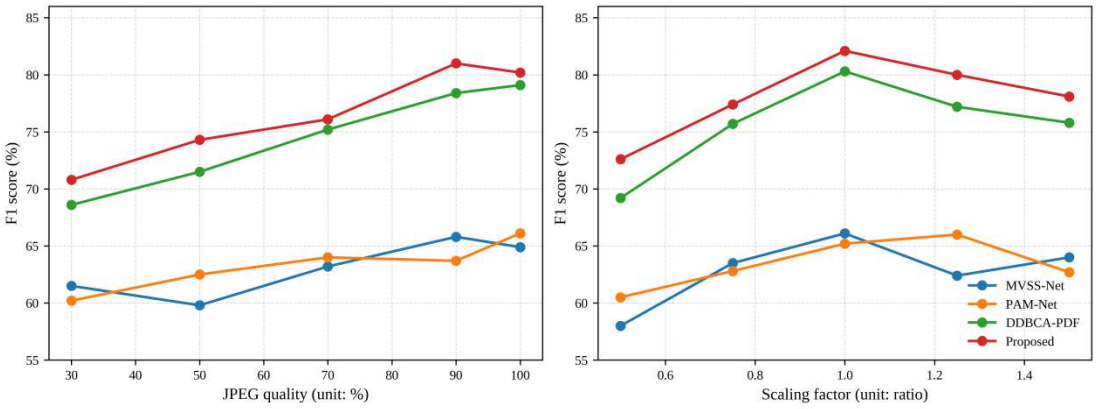


Figure 8 Robustness comparison under JPEG compression and image resizing

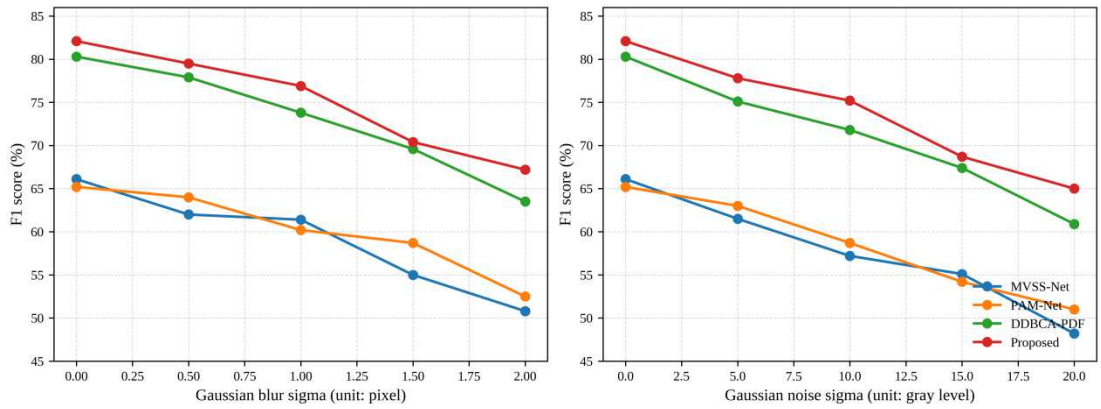


Figure 9 Robustness comparison under Gaussian blur and Gaussian noise

The Gaussian blur and Gaussian noise results further evaluate the model under low-quality image conditions. In Figure 9, the proposed method decreases from 82.1% F1 at blur sigma 0 to 67.2% F1 at blur sigma 2.0. DDBCA-PDF decreases from 80.3% to 63.5%, while MVSS-Net decreases from 66.1% to 50.8%. The proposed method therefore preserves a 3.7 percentage-point advantage over DDBCA-PDF under the strongest blur condition. Under Gaussian noise, the proposed method decreases from 82.1% F1 to 65.0% F1 when the noise level increases to 20 gray levels. DDBCA-PDF drops to 60.9%, and MVSS-Net drops to 48.2%. Although noise and blur both weaken local manipulation traces, the proposed method retains better localization ability, which suggests that pseudo-label supervision and boundary refinement can reduce the dependence on fragile low-level features.

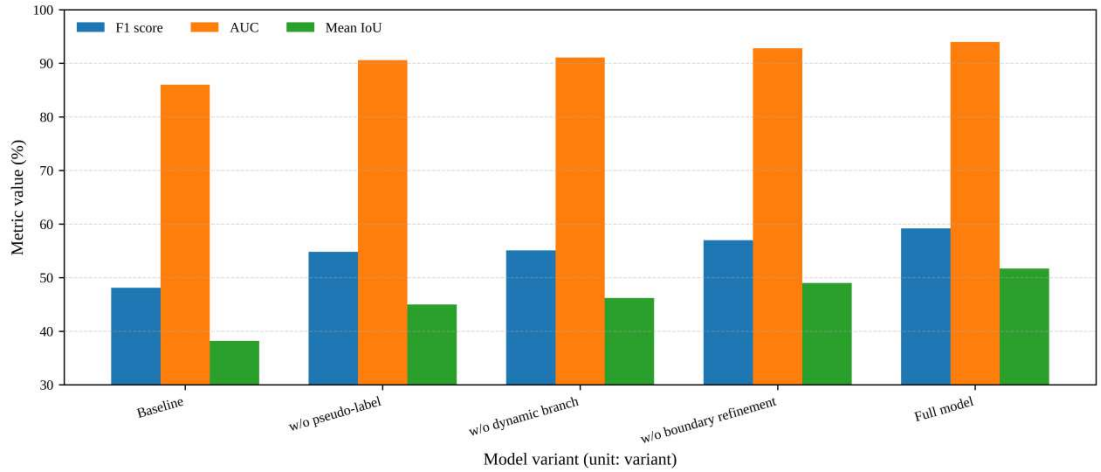


Figure 10 Ablation results of main modules

The ablation results in Figure 10 verify the contribution of each core component. The baseline model achieves 48.1% F1, 86.0% AUC, and 38.2% mean IoU. After removing

pseudo-label learning, the model variant obtains 54.8% F1, which still improves over the baseline but is 4.4 percentage points lower than the full model. Without the dynamic branch, the F1 score is 55.1%, and without boundary refinement, the F1 score is 57.0%. The full model achieves 59.2% F1, 94.0% AUC, and 51.7% mean IoU. Compared with the baseline, the complete framework improves F1 by 11.1 percentage points, AUC by 8.0 percentage points, and mean IoU by 13.5 percentage points. These results indicate that the three components are complementary: Pseudo-label learning helps copy-move region association, dynamic branches improve adaptation to different tampering types, and boundary refinement improves mask quality.

The parameter sensitivity analysis in Figure 11 gives further evidence for the selected settings in Table 4. For the pseudo-label threshold, the best F1 score appears at 0.70, reaching 59.2%. A lower threshold of 0.50 introduces more noisy pseudo labels and reduces F1 to 55.0%, while a higher threshold of 0.90 becomes too conservative and reduces F1 to 56.6%. For the number of dynamic branches, three branches provide the best result of 59.2% F1. Using only one branch gives 54.0% F1, while four branches slightly reduce the score to 58.1%, suggesting that excessive branches may introduce redundant responses. For boundary loss weight, the best value is 0.20. When the weight is increased to 0.40, F1 decreases to 57.1%, indicating that overly strong boundary constraints may weaken region-level consistency.

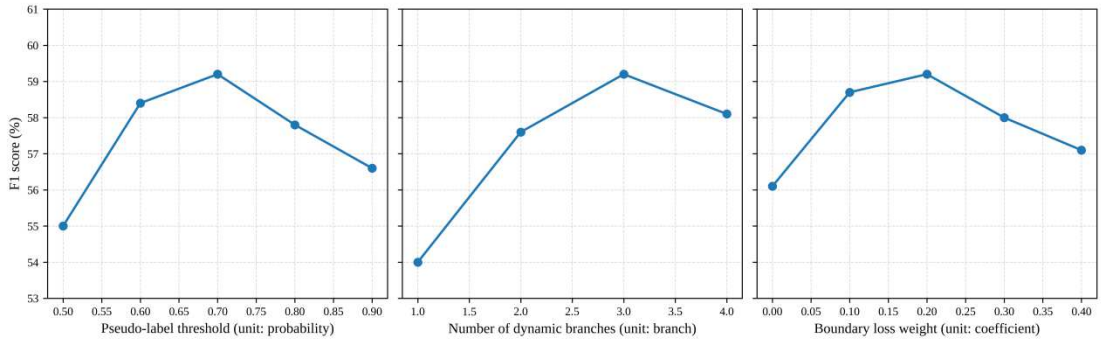


Figure 11 Influence analysis of pseudo-label quality, dynamic branch number, and boundary refinement loss

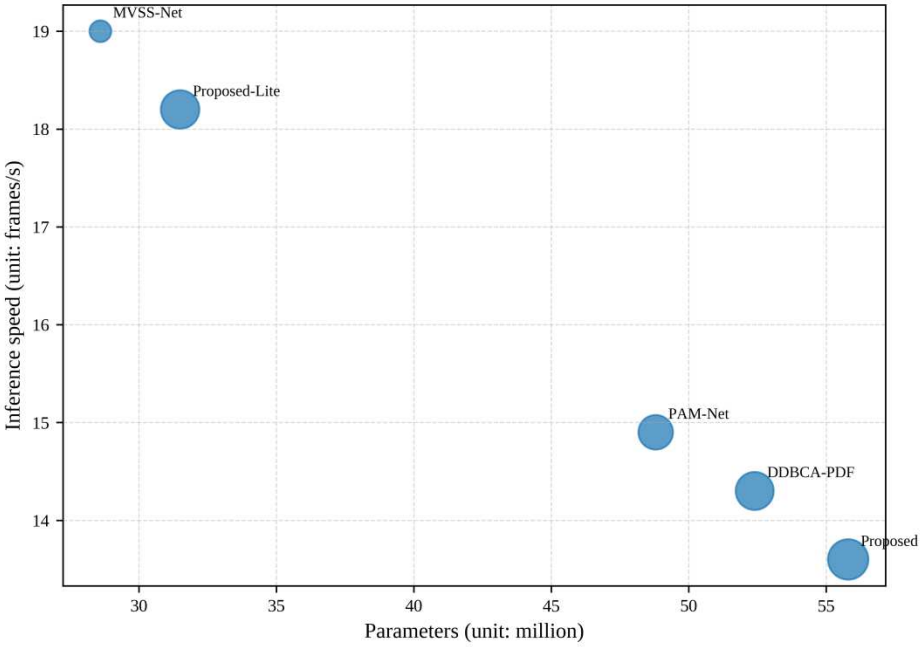


Figure 12 Comparison of inference speed, parameter size, and localization performance across different models

The complexity comparison in Figure 12 shows the trade-off between accuracy and efficiency. The proposed full model achieves the highest F1 score of 59.2%, but it also uses 55.8M parameters and runs at 13.6 frames/s. DDBCA-PDF uses 52.4M parameters and reaches 14.3 frames/s, but its F1 score is 56.4%, which is 2.8 percentage points lower than the proposed method. MVSS-Net is faster, with 19.0 frames/s and 28.6M parameters, but its F1 score is only 42.0%. Proposed-Lite provides a more balanced option, using 31.5M parameters and running at 18.2 frames/s while maintaining 57.0% F1. Compared with the full model, Proposed-Lite reduces parameters by 24.3M and increases speed by 4.6 frames/s, while losing only 2.2 percentage points in F1. Therefore, the full model is more suitable for accuracy-oriented forensic analysis, while Proposed-Lite is more suitable for practical scenarios requiring faster inference.

5 Conclusion

This paper proposes a pseudo-label-driven and tamper-type-aware dynamic multi-branch decoding framework for image tampering region localization, aiming to address key challenges such as heterogeneous manipulation traces, lack of source-region supervision in

copy-move forgeries, and poor sensitivity to small tampered regions. Unlike conventional binary segmentation approaches, the proposed method explicitly models manipulation diversity by integrating multi-scale feature encoding, pseudo-label-based source–target learning, and adaptive multi-branch decoding.

Specifically, the pseudo-label generation mechanism constructs source–target correspondence for copy-move forgeries using feature similarity and spatial exclusion constraints, while assigning confidence scores to suppress unreliable supervision. This allows the model to learn homologous relationships even without manual annotation of source regions. Meanwhile, the dynamic multi-branch decoder adaptively selects and fuses different decoding paths based on tampering type distribution and pseudo-label reliability, enabling more discriminative feature utilization for splicing, inpainting, and copy-move forgeries.

Experimental results validate the effectiveness of the proposed framework. In cross-dataset evaluation, the method achieves 75.4% F1 and 93.5% AUC on the COLUMBIA dataset, and 52.7% F1 and 87.2% AUC on the COVER dataset, outperforming existing methods such as DDBCA-PDF by up to 3.6% F1 improvement on copy-move localization. In intra-dataset experiments, the model reaches 99.0% F1 on COLUMBIA and 98.6% F1 on COVER, demonstrating strong fitting capability under consistent data distribution. Ablation studies further confirm the complementarity of each module, with the full model achieving 59.2% F1, 94.0% AUC, and 51.7% IoU, significantly improving over the baseline by 11.1%, 8.0%, and 13.5%, respectively.

Robustness analysis shows that the proposed method maintains stable performance under common image degradations, including JPEG compression (70.8% F1 at quality 30), Gaussian blur (67.2% F1 at σ=2.0), and Gaussian noise (65.0% F1 at noise level 20). These results demonstrate strong resistance to real-world distortions. Overall, the proposed framework effectively enhances localization accuracy, boundary consistency, and cross-manipulation generalization, providing a robust solution for practical image forensics applications.

References

[1] Farid, H. (2009). Image forgery detection. *IEEE Signal processing magazine*, *26*(2), 16-25.

[2] Birajdar, G. K., & Mankar, V. H. (2013). Digital image forgery detection using passive techniques: A survey. *Digital investigation*, *10*(3), 226-245.

[3] Castillo Camacho, I., & Wang, K. (2021). A comprehensive review of deep-learning-based methods for image forensics. *Journal of imaging*, *7*(4), 69.

[4] Verdoliva, L. (2020). Media forensics and deepfakes: an overview. *IEEE journal of selected topics in signal processing*, *14*(5), 910-932.

[5] Popescu, A. C., & Farid, H. (2005). Exposing digital forgeries by detecting traces of resampling. *IEEE Transactions on signal processing*, *53*(2), 758-767.

[6] Farid, H. (2009). Exposing digital forgeries from JPEG ghosts. *IEEE transactions on information forensics and security*, *4*(1), 154-160.

[7] Lukas, J., Fridrich, J., & Goljan, M. (2006). Digital camera identification from sensor pattern noise. *IEEE Transactions on Information Forensics and Security*, *1*(2), 205-214.

[8] Bianchi, T., & Piva, A. (2012). Image forgery localization via block-grained analysis of JPEG artifacts. *IEEE Transactions on Information Forensics and Security*, *7*(3), 1003-1017.

[9] Ferrara, P., Bianchi, T., De Rosa, A., & Piva, A. (2012). Image forgery localization via fine-grained analysis of CFA artifacts. *IEEE Transactions on Information Forensics and Security*, *7*(5), 1566-1577.

[10] Amerini, I., Ballan, L., Caldelli, R., Del Bimbo, A., & Serra, G. (2011). A sift-based forensic method for copy–move attack detection and transformation recovery. *IEEE transactions on information forensics and security*, *6*(3), 1099-1110.

[11] Christlein, V., Riess, C., Jordan, J., Riess, C., & Angelopoulou, E. (2012). An evaluation of popular copy-move forgery detection approaches. *IEEE Transactions on information forensics and security*, *7*(6), 1841-1854.

[12] Cozzolino, D., Poggi, G., & Verdoliva, L. (2015). Efficient dense-field copy–move forgery detection. *IEEE Transactions on Information Forensics and Security*, *10*(11), 2284-2297.

[13] Li, J., Li, X., Yang, B., & Sun, X. (2014). Segmentation-based image copy-move forgery detection scheme. *IEEE transactions on information forensics and security*, *10*(3), 507-518.

[14] Ardizzone, E., Bruno, A., & Mazzola, G. (2015). Copy–move forgery detection by matching triangles of keypoints. *IEEE Transactions on Information Forensics and*

Security, *10*(10), 2084-2094.

[15] Salloum, R., Ren, Y., & Kuo, C. C. J. (2018). Image splicing localization using a multi-task fully convolutional network (MFCN). *Journal of Visual Communication and Image Representation*, *51*, 201-209.

[16] Bappy, J. H., Simons, C., Nataraj, L., Manjunath, B. S., & Roy-Chowdhury, A. K. (2019). Hybrid LSTM and encoder–decoder architecture for detection of image forgeries. *IEEE transactions on image processing*, *28*(7), 3286-3300.

[17] Liu, Y., Zhu, X., Zhao, X., & Cao, Y. (2019). Adversarial learning for constrained image splicing detection and localization based on atrous convolution. *IEEE Transactions on Information Forensics and Security*, *14*(10), 2551-2566.

[18] Zhong, J. L., & Pun, C. M. (2019). An end-to-end dense-inceptionnet for image copy-move forgery detection. *IEEE Transactions on Information Forensics and Security*, *15*, 2134-2146.

[19] Weng, S., Zhu, T., Zhang, T., & Zhang, C. (2023). UCM-Net: A U-Net-like tampered-region-related framework for copy-move forgery detection. *IEEE Transactions on Multimedia*, *26*, 750-763.

[20] Li, H., Luo, W., Qiu, X., & Huang, J. (2017). Image forgery localization via integrating tampering possibility maps. *IEEE Transactions on Information Forensics and Security*, *12*(5), 1240-1252.

[21] Li, F., Pei, Z., Zhang, X., & Qin, C. (2022). Image manipulation localization using multi-scale feature fusion and adaptive edge supervision. *IEEE Transactions on Multimedia*, *25*, 7851-7866.

[22] Dong, C., Chen, X., Hu, R., Cao, J., & Li, X. (2022). Mvss-net: Multi-view multi-scale supervised networks for image manipulation detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*(3), 3539-3553.

[23] Kwon, M. J., Nam, S. H., Yu, I. J., Lee, H. K., & Kim, C. (2022). Learning jpeg compression artifacts for image manipulation detection and localization. *International Journal of Computer Vision*, *130*(8), 1875-1895.

[24] Gu, A. R., Nam, J. H., & Lee, S. C. (2022). FBI-Net: Frequency-based image forgery localization via multitask learning with self-attention. *IEEE access*, *10*, 62751-62762.

[25] Chen, B., Tan, W., Coatrieux, G., Zheng, Y., & Shi, Y. Q. (2020). A serial image copy-move forgery localization scheme with source/target distinguishment. *IEEE Transactions on Multimedia*, *23*, 3506-3517.

[26] Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431-3440).

[27] Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, *40*(4), 834-848.

[28] Barni, M., Phan, Q. T., & Tondi, B. (2020). Copy move source-target disambiguation through multi-branch CNNs. *IEEE Transactions on Information Forensics and Security*, *16*, 1825-1840.

[29] Hu, K., Chen, X., Chen, Z., Zhang, Y., & Gao, X. (2024). Multi-perspective pseudo-label generation and confidence-weighted training for semi-supervised semantic segmentation. *IEEE Transactions on Multimedia*, *27*, 300-311.

[30] Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7132-7141).

[31] Kaya, M., & Bilge, H. Ş. (2019). Deep metric learning: A survey. *Symmetry*, *11*(9), 1066.

[32] Schlemper, J., Oktay, O., Schaap, M., Heinrich, M., Kainz, B., Glocker, B., & Rueckert, D. (2019). Attention gated networks: Learning to leverage salient regions in medical images. *Medical image analysis*, *53*, 197-207.

[33] Taghanaki, S. A., Zheng, Y., Zhou, S. K., Georgescu, B., Sharma, P., Xu, D., ... & Hamarneh, G. (2019). Combo loss: Handling input and output imbalance in multi-organ segmentation. *Computerized Medical Imaging and Graphics*, *75*, 24-33.

[34] Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision* (pp. 2980-2988).

[35] Kervadec, H., Bouchtiba, J., Desrosiers, C., Granger, E., Dolz, J., & Ayed, I. B. (2019, May). Boundary loss for highly unbalanced segmentation. In *International conference on*

medical imaging with deep learning (pp. 285-296). PMLR.