DcDsDiff: Dual-Conditional and Dual-Stream Diffusion Model for Generative Image Tampering Localization

Qixian Hao[1] , **Shaozhang Niu**[1,2] , **Jiwei Zhang**[1,3*] and **Kai Wang**[1]
[1]Beijing Key Lab of Intelligent Telecommunication Software and Multimedia, Beijing University of Posts and Telecommunications, China
[2]Southeast Digital Economy Development Institute, China
[3]Key Laboratory of Interactive Technology and Experience System, Ministry of Culture and Tourism(BUPT), China
haoqixian123@163.com, {szniu, jwzhang666, kaiwang}@bupt.edu.cn

Abstract

Generative Image Tampering (GIT), due to its high diversity and realism, poses a significant challenge to traditional image tampering localization techniques. Consequently, this paper introduces a denoising diffusion probabilistic model-based DcDsDiff, which comprises a Dual-View Conditional Network (DVCN) and a Dual-Stream Denoising Network (DSDN). DVCN provides clues about the tampered areas. It extracts tampering features in the high-frequency view and integrates them with spatial domain features using attention mechanisms. DSDN jointly generates mask image and detail image, enhancing the generalization capability of the model against new tampering forms through iterative denoising. A multi-stream interaction mechanism enables the two generative tasks to promote each other, prompting the model to generate localization results that are rich in detail and complete. Experiments show that DcDsDiff outperforms mainstream methods in accurate localization, generalization, extensibility, and robustness. Code page: https://github.com/QixianHao/DcDsDiff-and-GIT10K.

1 Introduction

Diffusion-based image generation technology is widely noted for its superior quality in image creation. For example, Stable Diffusion's [Rombach *et al.*, 2022] inpainting function allows users to re-create specific parts of an image, enabling a rich variety of visual expressions and innovative effects. However, this technological advancement also presents challenges in information security. As image generation becomes more convincing, it's easier than ever to create deceptive image tampering. As shown in Figure 1, malicious actors could use these technologies to produce fake images, spread misinformation, and destabilize public opinion and social stability. Thus, detecting and localizing Generative Image Tampering (GIT) is becoming increasingly critical.

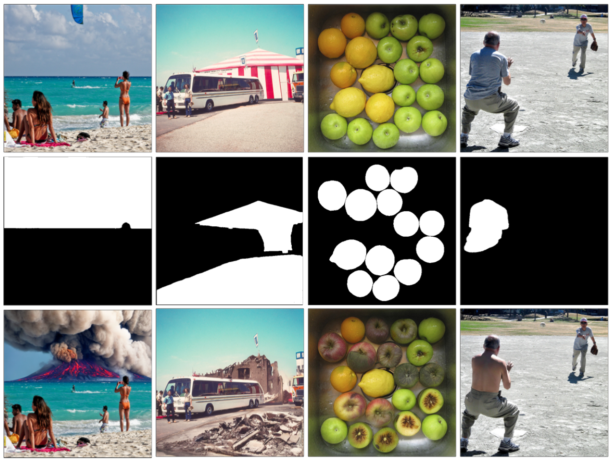

Figure 1: Examples of Generative Image Tampering (GIT). Top row shows real images, middle row shows ground truth (GT), and bottom row shows fake images. Column 1: calm sea to volcanic eruption; Column 2: intact building to ruins; Column 3: fresh fruit to rotten fruit; Column 4: clothed person to bare-chested.

Current mainstream Image Tampering Localization (ITL) methods, while effective in ITL tasks of Splicing, Copy-Move, or Removal, struggle with handling GIT. GIT is not limited to a single tampering form. Its diversity in purpose (e.g., removal, filling, replacing [Ju *et al.*, 2024]), generated object types, and mechanisms (e.g., example-guided [Yang *et al.*, 2023], context-aware [Zhuang *et al.*, 2025], text-guided [Ju *et al.*, 2024]) demands strong generalization capabilities of forensic models. Most ITL methods are built on conventional encoder-decoder architectures, following a linear information processing mechanism, lacking feedback loops or intermediate adjustments to optimize predictions. This fixed process lacks flexibility and cannot adapt to diverse tampering forms and complex scenes. Additionally, the generated objects in GIT are highly integrated with the background, making the edges of the tampered areas and nearby pixels highly uncertain. Conventional ITL methods often misidentify these edges due to over-confidence in linear processing, which hinders the generation of accurate and complete localization results.

[*]Corresponding Author

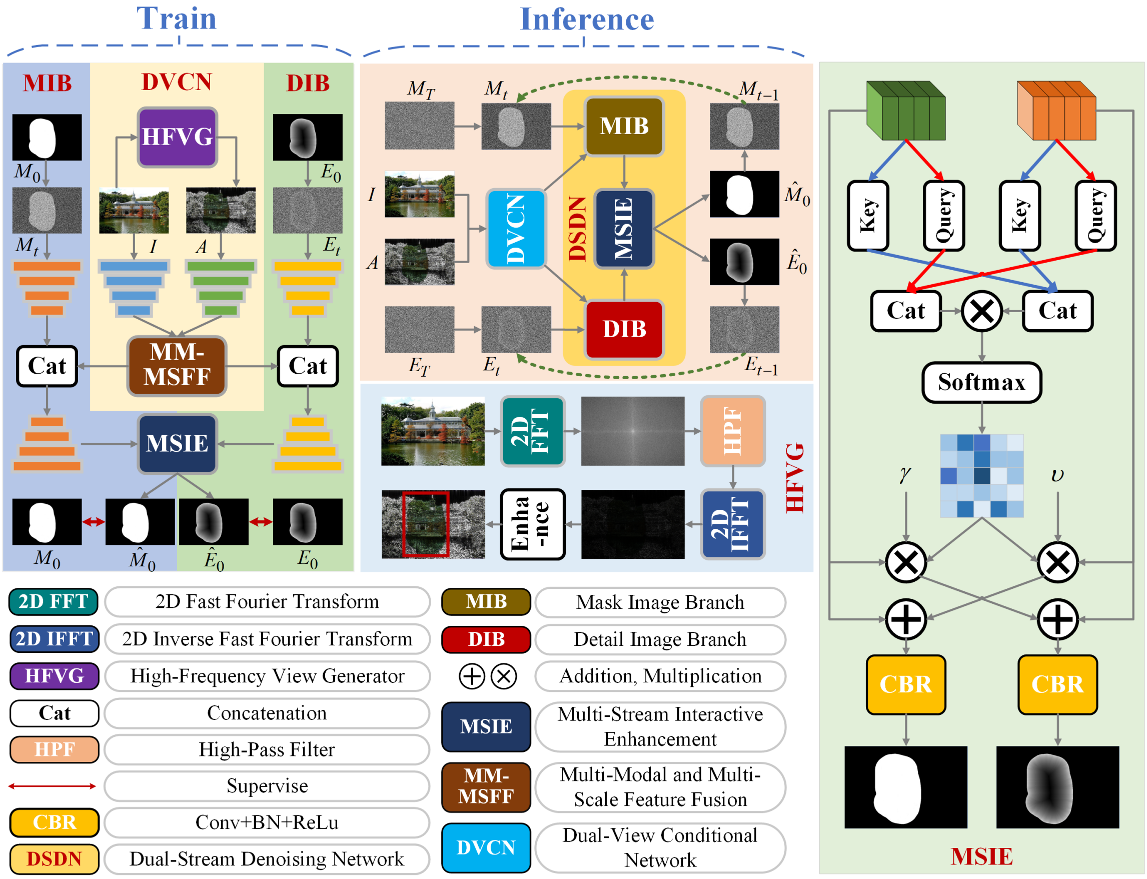

Figure 2: The structure of DcDsDiff, HFVG, and MSIE. DcDsDiff is composed of a DVCN and DSDN. The DVCN consists of two PVTv2 networks, one HFVG, and one MM-MSFF, while the DSDN consists of one Mask Image Branch (MIB), one Detail Image Branch (DIB), and one MSIE.

In response to the aforementioned issues, we propose a Dual-Conditional and Dual-Stream Diffusion Model (DcDsDiff), as shown in Figure 2. Specifically, we introduce a diffusion model strategy for Generative Image Tampering Localization (GITL) task, redefining GITL as a diffusion model-based bi-generation task, achieving joint generation of mask image (tampered area) and detail image (edge and surrounding pixel). Firstly, the model includes a parallel Dual-Stream Denoising Network (DSDN) that refines predictions through iterative denoising. This iterative process enhances DcDsDiff's understanding of data distributions, allowing it to better generalize to diverse tampering forms. It also enables the model to fine-tune predictions based on image conditions, thereby helping the model to make more cautious predictions in edge and detail areas, thus avoiding overconfidence. To achieve mutual promotion between the two streams, we introduced a Multi-Stream Interaction Enhancement (MSIE) module. This module uses detail features to guide the prediction of the mask image, improving the mask image stream's perception of detail areas. Simultaneously, it utilizes mask features to provide overall positional information for the detail image, thereby improving detail image's completeness. Secondly, DcDsDiff also includes a Dual-View Conditional Network (DVCN) that provides clues to the DSDN. The RGB conditional network processes spatial information, while the high-frequency conditional network captures tampering traces in the high-frequency view, which are invisible in the RGB view. This high-frequency view is generated by the High-Frequency View Generator (HFVG) to reflect local anomalies caused by GIT. Additionally, we designed a Multi-Modal and Multi-Scale Feature Fusion (MM-MSFF) module. This module aligns features of both modalities spatially, re-calibrates features at the channel level, integrates features of the same level based on their importance, combines features from different levels to provide the DSDN with the necessary fused conditional feature.

To sum up, our main contributions are as follows:

- A diffusion-based DcDsDiff model was proposed, introduced a DSDN to simultaneously generate mask image and detail image, and designed a MSIE to enhance information complementarity between the two streams.
- Designed a DVCN to extract tampering features from multiple views, introduced an HFVG to capture high-frequency anomalies, and designed a MM-MSFF to integrate multi-modal and multi-scale features.

- Extensive experiments demonstrate that DcDsDiff has significant advantages in terms of localization performance, generalization, extensibility, and robustness.

2 Related Work

2.1 ITL

We categorize mainstream ITL methods into four major categories: traditional methods [Ren *et al.*, 2023; Liu *et al.*, 2022], noise view-assisted methods [Niloy *et al.*, 2023; Dong *et al.*, 2022; Lin *et al.*, 2023; Wu *et al.*, 2019; Ji *et al.*, 2023; Wu and Zhou, 2021; Frick and Steinebach, 2024], frequency domain view-assisted methods [Frick and Steinebach, 2024; Wang *et al.*, 2022a; Liu *et al.*, 2023], and edge-assisted multi-task methods[Shi *et al.*, 2023; Ren *et al.*, 2024; Hao *et al.*, 2024c; Hao *et al.*, 2024a; Dong *et al.*, 2022; Wang *et al.*, 2025]. Traditional methods primarily rely on the attribute differences between tampered and untampered regions in spatial domain features. Noise view/frequency domain-assisted methods aim to detect tampering traces in noise view/frequency domain that are invisible in the RGB view. However, poorly designed noise view/frequency domain generators often introduce redundant information, leading to adverse interference in localization results. Edge-assisted methods aim to capture edge inconsistencies caused by tampering. However, GIT lacks distinct boundaries. An overly confident edge prior can often mislead the model into producing erroneous localization results.

2.2 Diffusion Models for Image Segmentation

Currently, the application of diffusion models in the field of image segmentation is relatively limited. BerDiff [Chen *et al.*, 2023] is a conditional Bernoulli diffusion model for medical image segmentation, generating accurate and diverse segmentation masks through Bernoulli noise and multiple samplings. CamoDiffusion [Chen *et al.*, 2024] utilizes the denoising mechanism to gradually reduce the differences between initial noise and the true mask, effectively improving the detection performance of camouflaged objects. Additionally, BiDiCOS [Jiang *et al.*, 2024] combines bilateral diffusion models with depth estimation techniques, aiming to optimize the segmentation process by fusing depth information, enhancing the accuracy and robustness of camouflaged object detection.

3 Proposed DcDsDiff

3.1 Background

In this paper, we propose a dual-stream diffusion model based on denoising diffusion probabilistic models, which generates mask image and detail image simultaneously. As shown in Figure 3, the diffusion model is mainly divided into two stages: the forward process and the backward process. In the forward process, we start with noise-free original mask image M_0 and original detail image E_0, iteratively adding Gaussian noise until the diffusion step T is large enough that the original mask image and detail image completely degrade into noise. This process can be described by the following Markov process:

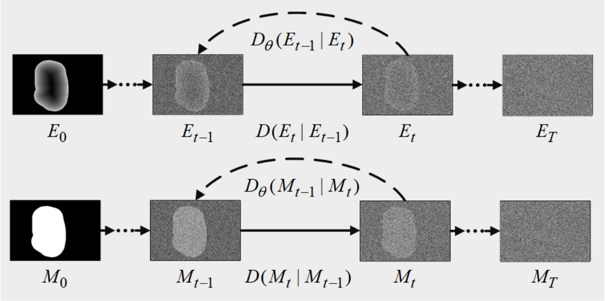

Figure 3: The diffusion process. From left to right is the forward process of adding noise, and from right to left is the backward denoising process.

$$D(M_t|M_{t-1}) = N(M_t; \sqrt{1-\beta_t} \cdot M_{t-1}, \beta_t \cdot \mathrm{I}), \quad (1)$$

$$D(E_t|E_{t-1}) = N(E_t; \sqrt{1-\beta_t} \cdot E_{t-1}, \beta_t \cdot \mathrm{I}), \quad (2)$$

where, t runs from 1 to T, and variance is controlled by noise schedule $\beta_t \in (0,1)$. I is an identity matrix. M_t and E_t represent the mask image and detail image at time step t, respectively, and can be obtained by the following equations:

$$D(M_t|M_0) = N(M_t; \sqrt{\overline{\alpha}_t} \cdot M_0, (1-\overline{\alpha}_t) \cdot \mathrm{I}), \quad (3)$$

$$D(E_t|E_0) = N(E_t; \sqrt{\overline{\alpha}_t} \cdot E_0, (1-\overline{\alpha}_t) \cdot \mathrm{I}), \quad (4)$$

where, $\overline{\alpha}_t = \prod_{i=1}^{T} \alpha_t$, $\alpha_t = 1 - \beta_t$. The backward process is the reverse of the forward process. The model starts from two noisy images, gradually denoises them, and eventually restores the clean original images.

$$D_\theta(M_{t-1}|M_{\mathrm{t}}) = N(M_{t-1}; \mu_\theta(M_t,t), \sum\nolimits_\theta (M_t,t), \quad (5)$$

$$D_\theta(E_{t-1}|E_{\mathrm{t}}) = N(E_{t-1}; \mu_\theta(E_t,t), \sum\nolimits_\theta (E_t,t)), \quad (6)$$

where, θ refers to the model parameters. $\mu_\theta(M_t,t)$ and $\mu_\theta(E_t,t)$ refer to the means predicted by the model at a given time step t. $\sum_\theta (M_t,t)$ and $\sum_\theta (E_t,t)$ refer to the predicted covariances. $\sum_\theta (M_t,t)$ and $\sum_\theta (E_t,t)$ are set to $\sigma_t^2 = \frac{1-\overline{a}_{t-1}}{1-\overline{a}_t}\beta_t$. $\mu_\theta(M_t,t)$ and $\mu_\theta(E_t,t)$ can be represented as:

$$\mu_\theta(M_t,t) = \frac{\sqrt{\alpha_t}(1-\overline{\alpha}_{t-1})}{1-\overline{\alpha}_t} M_t + \frac{\sqrt{\overline{\alpha}_{t-1}}\beta_t}{1-\overline{\alpha}_t} \hat{M}_0, \quad (7)$$

$$\mu_\theta(E_t,t) = \frac{\sqrt{\alpha_t}(1-\overline{\alpha}_{t-1})}{1-\overline{\alpha}_t} E_t + \frac{\sqrt{\overline{\alpha}_{t-1}}\beta_t}{1-\overline{\alpha}_t} \hat{E}_0, \quad (8)$$

where, $\hat{M}_0$ and $\hat{E}_0$ refer to the model's predicted mask image and detail image.

3.2 Overview

The architecture of DcDsDiff is shown in Figure 2. In training, the input are an RGB image I with the size of $W \times H \times 3$, the GT of the mask image M_0, and the GT of the detail image E_0. The output are the predicted mask image $\hat{M}_0$ and

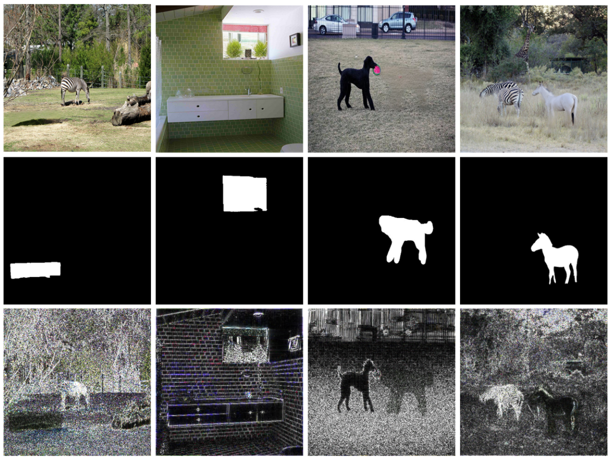

Figure 4: Tampering trace in high-frequency view. The first line shows fake image, the second line shows GT, and the third line shows high-frequency view.

detail image $\hat{E}_0$. Specifically, firstly, we use the HFVG to generate the high-frequency view of I, and then input I and high-frequency view A into the DVCN. In DVCN, we employ two pre-trained PVTv2 [Wang *et al.*, 2022b] models as the backbones. These two PVTv2 models can respectively generate spatial feature set r_1, r_2, r_3, r_4 and high-frequency feature set a_1, a_2, a_3, a_4 of different levels. We utilize the MM-MSFF to fuse the multi-modal and multi-scale features r_{1-4} and a_{1-4} to obtain the conditional feature c. Secondly, the mask image M_0 and the detail image E_0 undergo the noise addition process to generate noisy images M_t and E_t at time step t. Subsequently, we use the DSDN to denoise the noisy images M_t and E_t under the guidance of c, and leverage the MSIE to achieve information complementarity between the two streams. Finally, the losses of the predicted mask image $\hat{M}_0$ and detail image $\hat{E}_0$ are calculated. During the inference stage, the input is the GIT image, and the output is the localization results of the tampered areas. Specifically, firstly, we input the GIT image into the trained DcDsDiff model and repeat the operations of the DVCN from the training phase. Secondly, we initialize two Gaussian noise images M_T and E_T as inputs to the DSDN. Through iterative denoising by the DSDN, we gradually recover the detail image $\hat{E}_0$ and the mask image $\hat{M}_0$ from the noise images. Finally, we average the multiple predicted images generated during the iteration process to obtain the final predicted mask image $\hat{M}_0^f$ and detail image $\hat{E}_0^f$.

3.3 HFVG

As shown in Figure 4, GIT can cause local anomalies visible in the high-frequency view, which are difficult to capture in the RGB view. Therefore, we designed the HFVG to provide supplementary clues for RGB conditional network. The workflow of the HFVG is shown in Figure 2. Specifically, first, we perform a 2D Fast Fourier Transform (2D FFT) on the RGB image I along the channels. Then, we obtain the high-frequency components of the three channels through a high-pass filter. Next, we convert these high-frequency components back to the RGB domain by performing an inverse 2D Fast Fourier Transform (2D IFFT) and merging them. Finally, to further highlight the local anomalies in the high-frequency view, we enhance the values of this high-frequency view, resulting in the final high-frequency view A.

$$A = IFFT(HPF(FFT(I), \varsigma)) \otimes 10, \tag{9}$$

where, $\otimes$ refers to the multiplication operation. $FFT()$ and $IFFT()$ refer to the 2D FFT operation and the 2D IFFT operation, respectively. $HPF()$ denotes the high pass filter and ς is the manually designed threshold which controls the low frequency component to be filtered out.

3.4 MM-MSFF

The MM-MSFF consists of two parts: Multi-Modal Feature Fusion (MMFF) and Multi-Scale Feature Fusion (MSFF). MMFF aims to enhance and fuse spatial features and high-frequency features at the same level. Specifically, firstly, the positions of the tampered areas in the features of both modalities should be the same. Therefore, we use Spatial Attention (SA) to obtain their shared spatial attention map and align the features of both modalities using this spatial attention map.

$$\left\{ \begin{array}{l} att^{sa} = SA(r_i \otimes a_i) \\ r_i^{sa} = r_i \otimes att^{sa} \\ a_i^{sa} = a_i \otimes att^{sa} \end{array} \right. , i = 1, ..., 4. \tag{10}$$

where, $SA()$ refers to SA, which is an operation sequence composed of a Global Max Pooling operation (GMP) along the channel, a convolution operation with the kernel size of 3×3 (3×3 Conv), and a Sigmoid function [Liu *et al.*, 2021]. Secondly, high-frequency features mainly contain rich tampering features, while spatial features primarily contain rich texture and appearance information, with different focuses. Therefore, we use Channel Attention (CA) to further highlight the important information in both modalities.

$$\left\{ \begin{array}{l} r_i^{ca} = r_i \otimes CA(r_i^{sa}) \\ a_i^{ca} = a_i \otimes CA(a_i^{sa}) \end{array} \right. , i = 1, ..., 4. \tag{11}$$

where, $CA()$ refers to CA, which is an operation sequence composed of a GMP, a 1×1 Conv, a ReLU function, and a Sigmoid function. Lastly, the importance of features from the two modalities varies when processing different images. Ignoring their different contributions and directly fusing them can lead to a decrease in model performance. We consider dynamically allocating the weights of the two features when fusing them.

$$\left\{ \begin{array}{l} att_i^{fu} = \mathrm{Sig}(C_{3\times3}(r_i^{ca} \oplus a_i^{ca})) \\ r_i^{fu} = C_{3\times3}(r_i^{ca} \otimes att_i^{fu} \oplus r_i^{ca}) \\ a_i^{fu} = C_{3\times3}(a_i^{ca} \otimes att_i^{fu} \oplus a_i^{ca}) \\ f_i = C_{3\times3}(r_i^{fu} \oplus a_i^{fu}) \end{array} \right. , i = 1, ..., 4. \tag{12}$$

where, $\mathrm{Sig}()$ refers to Sigmoid function, $C_{3\times3}$ refers to 3×3 Standard convolution (3×3 CBR), $\oplus$ refers to the addition operation. High-level features contain richer semantic information, which is beneficial for the model to determine the approximate location of the tampered area. Low-level features contain more detailed information, which is beneficial for the model to generate more refined localization results. Therefore, in MSFF, we use a concatenation operation to fuse features from different levels to obtain $c = Cat(f_1, f_2, f_3, f_4)$, where, Cat refers to the concatenation operation. c will be input into the DSDN.

3.5 DSDN

In ITL tasks, accurately locating the edges of tampered areas is crucial, as they often reveal the overall contour and local details of the tampered area. However, GIT makes the tampered area blend seamlessly with the background, leaving minimal traces. Edge pixels are few and highly similar to surrounding pixels, making direct edge prediction challenging. In contrast, identifying detail areas of the tampered region is easier. These areas include not only edge pixels but also nearby pixels. The values of surrounding pixels, determined by distance transformation [Wei *et al.*, 2020b], are inversely proportional to the distance from the nearest edge, with higher values for pixels closer to the edge and lower values for those farther away. Compared to methods relying solely on edge pixels, detail maps include more pixels, facilitating a more balanced pixel distribution.

Traditionally, Denoising network aims to decode the denoised mask predictions $\hat{M}_0$ and $\hat{M}_{t-1}$ based on the diffusion paradigm. However, the DSDN aims to achieve joint prediction of the mask image and detail image through two parallel UNets (MIB and DIB). As shown in Figure 2, specifically, first, we use multiple convolution layers to encode the two input noisy images into two feature maps of the same size as the conditional feature c. Then, we fuse these two feature maps with c through a simple concatenation operation. Finally, we use multiple convolutions and up-sampling operations to gradually restore the size of the two feature maps and obtain the mask feature d^M and detail feature d^E. We adopted adaptive group normalization [Chen *et al.*, 2024] in both UNets, incorporating the time step information t into the convolutions, making our DSDN sensitive to changes in the time step.

MIB and DIB are not independent of each other; their interaction is of significant importance. The mask image stream provides semantic information about the tampered area, while the detail image stream provides contour and detail information of the tampered area. Effective interaction between the two can enable the model to generate localization results with rich detail information and high integrity. Inspired by [Qian *et al.*, 2020], we designed the MSIE module to implement a cross-attention mechanism, as shown in Figure 2. Specifically, we first extract the Query features and Key features of the mask features and detail features.

$$Q = Cat(C_{1\times1}(d^M), C_{1\times1}(d^E)), \tag{13}$$

$$K = Cat(C_{1\times1}(d^M)^*, C_{1\times1}(d^E)^*), \tag{14}$$

where, $*$ denotes the transpose operation. Then, we use the Query features and Key features to compute the weights.

$$att^{cs} = SoftMax(Q \otimes K), \tag{15}$$

where, $SoftMax()$ refers to SoftMax function. Finally, we use the cross-attention weights to enhance the features of both streams.

$$p^M = d^M \oplus C_{1\times1}(d^E \otimes att^{cs} \otimes \gamma), \tag{16}$$

$$p^E = d^E \oplus C_{1\times1}(d^M \otimes att^{cs} \otimes \upsilon), \tag{17}$$

where, γ and υ are learnable parameters used to adjust the intensity of attention. p^M and p^E will be used respectively to predict the mask image $\hat{M}_0$ and the detail image $\hat{E}_0$.

3.6 Loss Function

In terms of loss functions, we utilize a combination of Weighted Binary Cross-Entropy (WBCE) and Weighted Intersection over Union (WIoU) loss [Wei *et al.*, 2020a] for the predicted mask image $\hat{M}_0$. For the detail image $\hat{E}_0$, the average of L1 loss and L2 loss is applied.

$$\begin{array}{l} Loss_{total} = L_{WBCE+WIOU}(\hat{M}_0, M_0) \\ +0.5(L_{L1}(\hat{E}_0, E_0) + L_{L2}(\hat{E}_0, E_0)) \end{array}, \tag{18}$$

where, $L_{WBCE+WIOU}$ refers to the combination of WIoU loss and WBCE loss, L_{L1} refers to the L1 loss, and L_{L2} refers to the L2 loss.

4 Experiments

4.1 Experiment Settings

In this paper, we implemented DcDsDiff using the PyTorch framework. All experiments were conducted on a single NVIDIA GeForce RTX 4090 GPU. Input images were resized to 352×352 and augmented via random horizontal flipping. During the training phase, we used the AdamW optimizer with the learning rate of 0.001 and the batch size of 6. ς was set to 0.5. We employed a Signal-to-Noise Ratio (SNR)-based variance schedule [Hoogeboom *et al.*, 2023] to adjust the SNR of the diffusion process. The model was trained for 100 epochs. During the inference phase, the model undergoes ten iterative steps. For each step of sampling, we used a nonlinear method to select the value of t [Jiang *et al.*, 2024]. In terms of datasets, firstly, we constructed a GIT10K dataset containing 10,000 images using four common diffusion-based local inpainting methods: Brush Net (BN) [Ju *et al.*, 2024], Paint by Example (PE) [Yang *et al.*, 2023], Inpaint Anything (IA) [Yu *et al.*, 2023], and Power Paint (PP) [Zhuang *et al.*, 2025], with each method contributing 2,500 images. Secondly, to comprehensively evaluate DcDsDiff, we tested its performance on datasets containing other tampering types, including RLS [Hao *et al.*, 2024c], IMD [Novozamsky *et al.*, 2020], Nist16 [Guan *et al.*, 2016], DEFACTO Splicing (DEF) [Mahfoudi *et al.*, 2019], and AutoSplice (AUTO) [Jia *et al.*, 2023]. Finally, we divided the train and test sets of the aforementioned datasets in a 9:1 ratio. For evaluation metrics, we chose the F1-Score (F1) and Intersection over Union (IoU) to assess the performance of DcDsDiff.

4.2 Ablation Study

In this experiment, we conducted a detailed ablation study on DcDsDiff using the GIT10K dataset, which includes four sub-test sets: BN, PE, IA, and PP, each containing 250

Methods	Components	Test sets							
		BN		PE		IA		PP	
		F1	IoU	F1	IoU	F1	IoU	F1	IoU
Scheme1	Base	0.825	0.735	0.756	0.689	0.721	0.625	0.876	0.822
Scheme2	Base+DSDN	0.833	0.749	0.782	0.723	0.730	0.641	0.892	0.849
Scheme3	Base+DSDN+HFVG	0.843	0.761	0.819	0.775	0.773	0.679	0.914	0.877
Scheme4	Base+DSDN+HFVG+MM-MSFF	**0.866**	**0.782**	**0.854**	**0.803**	**0.818**	**0.738**	**0.919**	**0.882**

Table 1: The results of the ablation study. The train data is the mixed train set formed by the BN, PE, IA, and PP sub-train sets of GIT10K, totaling 9,000 tampered images. ”Base” refers to a minimal setup: a single U-Net for mask prediction using RGB-view PVTv2 features. The best result per sub-test set is highlighted in bold font.

Models	Test sets							
	BN		PE		IA		PP	
	F1	IoU	F1	IoU	F1	IoU	F1	IoU
MVSS-Net	0.66	0.56	0.38	0.34	0.36	0.27	0.67	0.61
MFI-Net	0.81	0.72	0.70	0.64	0.57	0.49	0.83	0.79
TA-Net	0.79	0.72	0.70	0.65	0.31	0.24	0.82	0.78
CFL-Net	0.70	0.60	0.42	0.37	0.41	0.33	0.67	0.61
EMF-Net	0.85	0.77	0.78	0.73	0.73	0.65	0.89	0.85
EC-Net	0.85	0.76	0.78	0.73	0.68	0.61	0.89	0.84
UGEE-Net	0.83	0.75	0.73	0.68	0.63	0.57	0.87	0.81
DcDsDiff	**0.87**	**0.78**	**0.85**	**0.80**	**0.82**	**0.74**	**0.92**	**0.88**

Table 2: The results of eight SOTA methods on GIT10K. The train data is the mixed train set of GIT10K. The best result per sub-test set is highlighted in bold font.

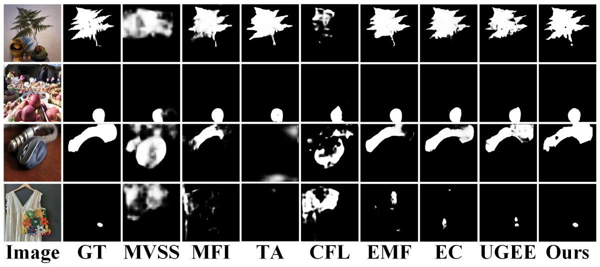

Figure 5: The localization results of eight methods. The fake images in the first to fourth rows are from the BN, PE, IA, and PP sub-test sets, respectively.

tampered images. We designed four experimental schemes, with their configurations and results detailed in Table 1. Firstly, the introduction of the DIB and MSIE significantly enhanced DcDsDiff’s performance, particularly for the PE sub-test set. Secondly, incorporating HFVG further improved performance based on Scheme 2, especially for the PE, IA, and PP sub-test sets, highlighting the importance of capturing tampering traces in the high-frequency view. Lastly, the introduction of MM-MSFF effectively fused the features of high-frequency view and RGB view. Compared to Scheme 3, Scheme 4 demonstrated notable improvements in F1 and IoU on BN, PE, and IA sub-test sets.

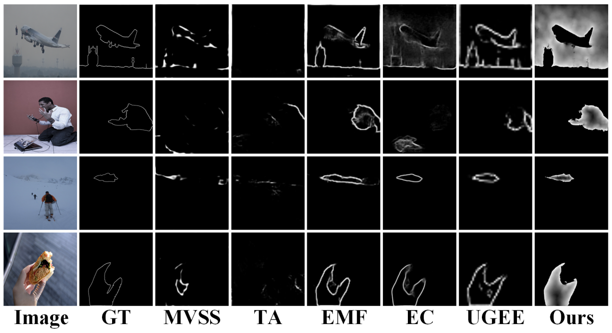

Figure 6: Edge images predicted by three edge-assisted methods and detail images predicted by DcDsDiff.

4.3 Comparison with State-of-the-Art Methods

In this experiment, we compared DcDsDiff with five ITL methods (MVSS-Net [Dong *et al.*, 2022], MFI-Net [Ren *et al.*, 2023], TA-Net [Shi *et al.*, 2023], CFL-Net [Niloy *et al.*, 2023], EMF-Net [Ren *et al.*, 2024]), UGEE-Net [Hao *et al.*, 2024b] and EC-Net [Hao *et al.*, 2024c] using the GIT10K dataset, with results shown in Table 2. The results indicate that DcDsDiff outperformed these ITL methods across all test sets. Notably, on the PE and IA test sets, DcDsDiff significantly exceeded the performance of the second-ranked EMF-Net, underscoring its superior localization capabilities. Figure 5 illustrates the localization results of these eight methods. Compared to the other seven methods, DcDsDiff provides more complete and detailed localization results. This advantage is primarily due to our detail image generation stream. Figure 6 illustrates the detail images generated by DcDsDiff and the edge images produced by five edge-assisted methods. As shown in Figure 6, when processing tampered areas that blend into the background, MVSS-Net and TA-Net struggle to generate complete edge images, while EMF-Net, EC-Net and UGEE-Net incorrectly identify non-edge pixels as edge pixels due to overconfidence. In contrast, DcDsDiff successfully avoids these issues, further validating the effectiveness of the detail image generation stream. More visualizations can be found in the supplementary material.

4.4 Cross-Generator Generalization Capability

In this experiment, we adopted a cross-dataset testing strategy to evaluate the generalization capabilities of DcDsDiff

Train sets	BN			PE			IA			PP		
Test sets	PE	IA	PP	BN	IA	PP	BN	PE	PP	BN	PE	IA
MVSS-Net	0.016	0.052	0.077	0.063	0.016	0.181	0.268	0.028	0.089	0.058	0.025	0.073
MFI-Net	0.261	0.256	0.340	0.162	0.120	0.524	0.285	0.032	0.088	0.113	0.122	0.110
TA-Net	0.248	0.239	0.327	0.094	0.158	0.435	0.172	0.028	0.071	0.087	0.151	**0.140**
CFL-Net	0.013	0.045	0.062	0.091	0.066	0.181	0.287	0.031	0.087	0.057	0.019	0.065
EMF-Net	0.379	0.264	0.471	0.132	0.092	0.552	0.334	0.022	0.098	0.043	0.098	0.096
DcDsDiff	**0.646**	**0.575**	**0.745**	**0.439**	**0.341**	**0.798**	**0.566**	**0.389**	**0.476**	**0.156**	**0.385**	0.053

Table 3: The cross-dataset testing results of the six methods. The best result per sub-test set is highlighted in bold font. F1 is used as the evaluation metric.

Baseline models	Test sets				
	RLS	IMD	Nist16	DEF	AUTO
MVSS-Net	0.434	0.363	0.879	0.715	0.913
MFI-Net	0.561	0.457	0.866	0.844	0.922
TA-Net	0.513	0.433	0.883	0.831	0.934
CFL-Net	0.33	0.433	0.884	**0.907**	0.917
EMF-Net	0.519	0.453	0.900	0.842	0.929
DcDsDiff	0.583	**0.495**	**0.936**	0.869	**0.949**

Table 4: The results of the six methods on the RLS, IMD, NIST16, DEF, and AUTO datasets. The best result per sub-test set is highlighted in bold font. F1 is used as the evaluation metric.

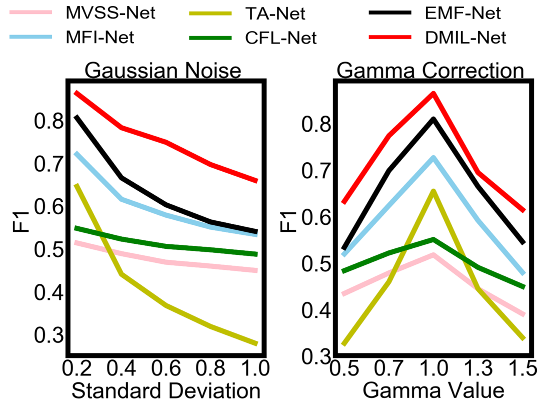

Figure 7: The results of robustness evaluation. The train data is the mixed train set of GIT10K. F1 is used as the evaluation metric.

and five other methods. The models were trained on the sub-train sets generated by specific GIT methods and tested on the sub-test sets generated by other GIT methods. The results are shown in Table 3. As seen in Table 3, DcDsDiff's performance is generally much better than the other five methods. The iterative denoising strategy of DcDsDiff enhances the model's understanding of data distributions, reduces over-reliance on labeled data, and improves the model's generalization capability.

4.5 Extensibility Evaluation

In this experiment, we evaluated the extensibility of DcDsDiff using the RLS, IMD, Nist16, DEF, and AUTO datasets. IMD consists of real-life manipulated images. RLS and Nist16 include three types of tampering: Splicing, Copy-Move, and Removal. DEF contains synthetic splicing images. And AUTO is an AIGC dataset generated by the DALL-E2 model. The results are presented in Table 4. All models are separately trained and tested on each dataset without cross-datasetmixing. Although DcDsDiff performs slightly worse than CFL-Net on the DEF test images, it significantly outperforms the other five methods on the RLS, IMD, Nist16, and AUTO test sets. This experiment demonstrated that DcDsDiff is a versatile ITL model.

4.6 Robustness Evaluation

In this experiment, we individually applied Gaussian noise and gamma correction attacks to the mixed test set of BN, PE, IA, and PP sub-test sets to assess the stability of DcDsDiff. We then compared its performance with five mainstream methods. These models were trained on the mixed train set of GIT10K. As illustrated in Figure 7, Gaussian noise does indeed interfere with GITL. As the intensity of the standard deviation increases, the performance of all models shows a significant decline. Compared to the other five models, DcDsDiff consistently demonstrates the best performance. This proves that DcDsDiff has a certain degree of robustness against Gaussian noise. Gamma correction also affects the model's localization performance. When the gamma value is below or above 1.0, all models show varying degrees of performance degradation. However, DcDsDiff still exhibits the best performance.

5 Conclusion

DcDsDiff demonstrates superiority in several aspects: Firstly, it introduces the strategy of DSDN to synchronously generate mask images and detail images, enhancing the model's generalization capability. Secondly, through MSIE, it achieves information complementarity, generating localization results with rich details. Thirdly, DVCN and HFVG effectively capture tampering features, while the MM-MSFF module strengthens feature fusion. Extensive experiments have proven the superiority of DcDsDiff compared to traditional ITL methods.

Acknowledgments

This work is supported by National Key Research and Development Program of China [grant number 2024YFF0907404].

References

[Chen *et al.*, 2023] Tao Chen, Chenhui Wang, and Hongming Shan. Berdiff: Conditional bernoulli diffusion model for medical image segmentation. In *International conference on medical image computing and computer-assisted intervention*, pages 491–501. Springer, 2023.

[Chen *et al.*, 2024] Zhongxi Chen, Ke Sun, and Xianming Lin. Camodiffusion: Camouflaged object detection via conditional diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 1272–1280, 2024.

[Dong *et al.*, 2022] Chengbo Dong, Xinru Chen, Ruohan Hu, Juan Cao, and Xirong Li. Mvss-net: Multi-view multi-scale supervised networks for image manipulation detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3539–3553, 2022.

[Frick and Steinebach, 2024] Raphael Antonius Frick and Martin Steinebach. Diffseg: Towards detecting diffusion-based inpainting attacks using multi-feature segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3802–3808, 2024.

[Guan *et al.*, 2016] HY Guan, YY Lee, A Yates, A Delgado, D Zhou, D Joy, and A Pereira. Nist nimble 2016 datasets, 2016.

[Hao *et al.*, 2024a] Qixian Hao, Ruyong Ren, Shaozhang Niu, Kai Wang, Maosen Wang, and Jiwei Zhang. Ugee-net: Uncertainty-guided and edge-enhanced network for image splicing localization. *Neural Networks*, page 106430, 2024.

[Hao *et al.*, 2024b] Qixian Hao, Ruyong Ren, Shaozhang Niu, Kai Wang, Maosen Wang, and Jiwei Zhang. Ugee-net: Uncertainty-guided and edge-enhanced network for image splicing localization. *Neural Networks*, 178:106430, 2024.

[Hao *et al.*, 2024c] Qixian Hao, Ruyong Ren, Kai Wang, Shaozhang Niu, Jiwei Zhang, and Maosen Wang. Ec-net: General image tampering localization network based on edge distribution guidance and contrastive learning. *Knowledge-Based Systems*, 293:111656, 2024.

[Hoogeboom *et al.*, 2023] Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In *International Conference on Machine Learning*, pages 13213–13232. PMLR, 2023.

[Ji *et al.*, 2023] Kaixiang Ji, Feng Chen, Xin Guo, Yadong Xu, Jian Wang, and Jingdong Chen. Uncertainty-guided learning for improving image manipulation detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22456–22465, 2023.

[Jia *et al.*, 2023] Shan Jia, Mingzhen Huang, Zhou Zhou, Yan Ju, Jialing Cai, and Siwei Lyu. Autosplice: A text-prompt manipulated image dataset for media forensics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 893–903, 2023.

[Jiang *et al.*, 2024] Xinhao Jiang, Wei Cai, Yao Ding, Xin Wang, Danfeng Hong, Xingyu Di, and Weijie Gao. Bidicos: Camouflaged object segmentation via bilateral diffusion model. *Expert Systems with Applications*, 255:124747, 2024.

[Ju *et al.*, 2024] Xuan Ju, Xian Liu, Xintao Wang, Yuxuan Bian, Ying Shan, and Qiang Xu. Brushnet: A plug-and-play image inpainting model with decomposed dual-branch diffusion. *arXiv preprint arXiv:2403.06976*, 2024.

[Lin *et al.*, 2023] Xun Lin, Shuai Wang, Jiahao Deng, Ying Fu, Xiao Bai, Xinlei Chen, Xiaolei Qu, and Wenzhong Tang. Image manipulation detection by multiple tampering traces and edge artifact enhancement. *Pattern Recognition*, 133:109026, 2023.

[Liu *et al.*, 2021] Zhengyi Liu, Yacheng Tan, Qian He, and Yun Xiao. Swinnet: Swin transformer drives edge-aware rgb-d and rgb-t salient object detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(7):4486–4497, 2021.

[Liu *et al.*, 2022] Xiaohong Liu, Yaojie Liu, Jun Chen, and Xiaoming Liu. Pscc-net: Progressive spatio-channel correlation network for image manipulation detection and localization. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(11):7505–7517, 2022.

[Liu *et al.*, 2023] Weihuang Liu, Xi Shen, Chi-Man Pun, and Xiaodong Cun. Explicit visual prompting for low-level structure segmentations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19434–19445, 2023.

[Mahfoudi *et al.*, 2019] Gaël Mahfoudi, Badr Tajini, Florent Retraint, Frederic Morain-Nicolier, Jean Luc Dugelay, and PIC Marc. Defacto: Image and face manipulation dataset. In *2019 27Th european signal processing conference (EUSIPCO)*, pages 1–5. IEEE, 2019.

[Niloy *et al.*, 2023] Fahim Faisal Niloy, Kishor Kumar Bhaumik, and Simon S Woo. Cfl-net: image forgery localization using contrastive learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4642–4651, 2023.

[Novozamsky *et al.*, 2020] Adam Novozamsky, Babak Mahdian, and Stanislav Saic. Imd2020: A large-scale annotated dataset tailored for detecting manipulated images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops*, pages 71–80, 2020.

[Qian *et al.*, 2020] Yuyang Qian, Guojun Yin, Lu Sheng, Zixuan Chen, and Jing Shao. Thinking in frequency: Face forgery detection by mining frequency-aware clues. In *European conference on computer vision*, pages 86–103. Springer, 2020.

[Ren *et al.*, 2023] Ruyong Ren, Qixian Hao, Shaozhang Niu, Keyang Xiong, Jiwei Zhang, and Maosen Wang. Mfi-net: Multi-feature fusion identification networks for artificial intelligence manipulation. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(2):1266–1280, 2023.

[Ren *et al.*, 2024] Ruyong Ren, Qixian Hao, Feng Gu, Shaozhang Niu, Jiwei Zhang, and Maosen Wang. Emf-net: An edge-guided multi-feature fusion network for text manipulation detection. *Expert Systems with Applications*, 249:123548, 2024.

[Rombach *et al.*, 2022] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[Shi *et al.*, 2023] Zenan Shi, Haipeng Chen, and Dong Zhang. Transformer-auxiliary neural networks for image manipulation localization by operator inductions. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(9):4907–4920, 2023.

[Wang *et al.*, 2022a] Junke Wang, Zuxuan Wu, Jingjing Chen, Xintong Han, Abhinav Shrivastava, Ser-Nam Lim, and Yu-Gang Jiang. Objectformer for image manipulation detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2364–2373, 2022.

[Wang *et al.*, 2022b] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):415–424, 2022.

[Wang *et al.*, 2025] Kai Wang, Shaozhang Niu, Qixian Hao, and Jiwei Zhang. Inpdiffusion: Image inpainting localization via conditional diffusion models. *arXiv preprint arXiv:2501.02816*, 2025.

[Wei *et al.*, 2020a] Jun Wei, Shuhui Wang, and Qingming Huang. F^{3}net: fusion, feedback and focus for salient object detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 12321–12328, 2020.

[Wei *et al.*, 2020b] Jun Wei, Shuhui Wang, Zhe Wu, Chi Su, Qingming Huang, and Qi Tian. Label decoupling framework for salient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13025–13034, 2020.

[Wu and Zhou, 2021] Haiwei Wu and Jiantao Zhou. Iid-net: Image inpainting detection network via neural architecture search and attention. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(3):1172–1185, 2021.

[Wu *et al.*, 2019] Yue Wu, Wael AbdAlmageed, and Premkumar Natarajan. Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9543–9552, 2019.

[Yang *et al.*, 2023] Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by example: Exemplar-based image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18381–18391, 2023.

[Yu *et al.*, 2023] Tao Yu, Runseng Feng, Ruoyu Feng, Jinming Liu, Xin Jin, Wenjun Zeng, and Zhibo Chen. Inpaint anything: Segment anything meets image inpainting. *arXiv preprint arXiv:2304.06790*, 2023.

[Zhuang *et al.*, 2025] Junhao Zhuang, Yanhong Zeng, Wenran Liu, Chun Yuan, and Kai Chen. A task is worth one word: Learning with task prompts for high-quality versatile image inpainting. In *European Conference on Computer Vision*, pages 195–211. Springer, 2025.