DiffForensics: Leveraging Diffusion Prior to Image Forgery Detection and Localization

Zeqin Yu*,[1] Jiangqun Ni†,[2,3] Yuzhen Lin*,[4] Haoyi Deng[4] Bin Li†,[4]
[1]School of Computer Science and Engineering, Sun Yat-sen University
[2]School of Cyber Science and Technology, Sun Yat-sen University
[3]Department of New Networks, Peng Cheng Laboratory
[4]Guangdong Key Laboratory of Intelligent Information Processing, Shenzhen University

Abstract

As manipulating images may lead to misinterpretation of the visual content, addressing the image forgery detection and localization (IFDL) problem has drawn serious public concerns. In this work, we propose a simple assumption that the effective forensic method should focus on the mesoscopic properties of images. Base on the assumption, a novel two-stage self-supervised framework leveraging the diffusion model for IFDL task, i.e., *DiffForensics, is proposed in this paper. The DiffForensics begins with self-supervised denoising diffusion paradigm equipped with the module of encoder-decoder structure, by freezing the pre-trained encoder* (e.g., *in ADE-20K) to inherit macroscopic features for general image characteristics, while encouraging the decoder to learn microscopic feature representation of images, enforcing the whole model to focus the mesoscopic representations. The pre-trained model as a prior, is then further fine-tuned for IFDL task with the customized Edge Cue Enhancement Module (ECEM), which progressively highlights the boundary features within the manipulated regions, thereby refining tampered area localization with better precision. Extensive experiments on several public challenging datasets demonstrate the effectiveness of the proposed method compared with other state-of-the-art methods. The proposed DiffForensics could significantly improve the model's capabilities for both accurate tamper detection and precise tamper localization while concurrently elevating its generalization and robustness.*

1. Introduction

Manipulating images has become increasingly effortless as rapid advances in image editing tools, such as GAN [24, 25] and diffusion models [2, 36]. Users can easily forge on-the-fly images that do not exist or realize. The risks posed by such forged images in the wrong hands are obvious in terms of politics, economics, and personal privacy. Accordingly, the countermeasures being desired to identify image forgery have become an urgent topic in social security.

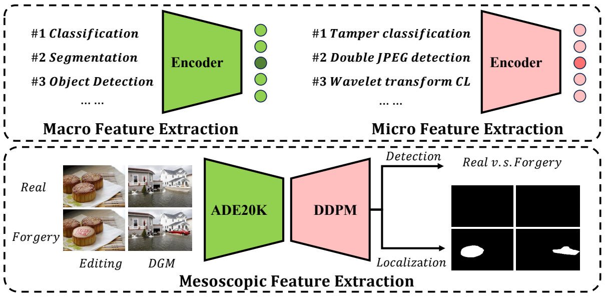


Figure 1. Comparison of the proposed pipeline (lower) with the conventional one (upper). Our approach could effectively integrate the macroscopic features with the microscopic features so that the model concentrates on the mesoscopic properties of tampered images and achieves better IFDL performance in both Artificial Editing and Deep Generative Model (DGM) images.

To push the frontier of image forensics, in this work, we study the image forgery detection and localization (IFDL) task, particularly partial modifications that change the image semantics. In general, the IFDL task involves binary classification (authentic versus forgery) at both the image level (detection) and the pixel level (localization). Until now, the state-of-the-arts [8, 16, 17, 22, 30, 31, 42, 45, 46] are commonly built upon the deep learning-based semantic segmentation meta-framework, consisting of two components, *i.e.*, the encoder and decoder. The encoder extracts the image features, subsequently processed by the decoder to predict classification results and forgery masks. Despite considerable advances in the area, current SOTA detectors are not yet performant enough for in-the-wild deployment, mainly due to their shortfall in generalization, robustness, and detection performance.

*Equal contribution. †Corresponding author.

Inspired by MesoNet [1], we propose to address the IFDL problem by focusing on the *mesoscopic* properties of images. Indeed, *microscopic* analyses based on artifacts (*e.g.*, image noise) cannot be applied in a social media laundering context because the post-processing will inevitably weaken forensic traces. Similarly, at a higher semantic level (*i.e.*, *macroscopic*), the human eye struggles to distinguish forged images. That is why we propose to adopt an intermediate approach.

To achieve this goal, we propose DiffForensics, a novel two-stage self-supervised method for the IFDL task. The training process begins with self-supervised denoising diffusion pretraining stage followed by a multi-task fine-tuning stage for IFDL. In the first stage, we freeze the encoder which is pre-trained with segmentation task (*e.g.*, ADE20K) [44] to retain the ability to extract macroscopic semantic features, while encouraging the decoder to learn the microscopic features relevant to forgery images with the self-supervised denoising diffusion paradigm. By integrating above schemes for training the encoder and decoder, which are respectively concentrating on macroscopic and microscopic features, we obtain the model that can learn the representation with mesoscopic features. In the second stage, we then fine-tune the pre-trained model (both the encoder and decoder) with the supervision of forgery images in the second stage. We propose a edge cue enhancement module (ECEM) and integrate it into the decoder across multiple scales, which aims to highlight the traces of tampered regions from coarse to fine. Extensive experiments demonstrate that our method outperforms several state-of-the-art competitors on several public datasets in terms of generalization and robustness performances.

The main contributions of this paper are summarized as follows:

- We propose a two-stage learning framework for IFDL tasks combining macro-features and micro-features, which consists of a self-supervised denoising diffusion pre-training stage and a multi-task fine-tuning stage. To the best of our knowledge, it is the first work to explore the denoising diffusion paradigm for the IFDL task.
- We propose a novel edge cue enhancement module, which is integrated into the decoder across multiple scales for enhancing tampered edge traces from coarse to fine.
- Extensive experimental results demonstrate that our proposed method achieves superior performances compared to state-of-the-art competitors on several recently emerged datasets, including both artificially manipulated and AI-generated images.

2. Related Work

Denoising Diffusion Probabilistic Models. The denoising diffusion probability model (DDPM) mainly consists of two stages [19], *i.e.*, the diffusion process that progressively adds random noise to data, and the reverse process that learn to reconstruct the desired data samples from the noise. In addition to being widely used in generative models [11] such as image generation [13, 33, 35, 38], image inpainting [10, 36], and image editing [2, 9], its potential representation learning ability has also found applications in other computer vision tasks, such as image segmentation [4, 6] and anomaly detection [41, 43]. By executing the noise estimation and reconstruction process, the denoising diffusion paradigm can effectively learn the microscopic noise pattern of the image. Meanwhile, noise analysis is one of the powerful solutions for the IFDL task. Thus, it makes sense to introduce the denoising diffusion paradigm for the IFDL task.

Methods	**Task**		**Model weight**	
	Loc.	**Det.**	**Encoder**	**Decoder**
H-LSTM [3]	✓	×	ImageNet	-
Mantra-Net [42]	✓	×	**Tamper Cls**	-
HP-FCN [30]	✓	×	ImageNet	-
SPAN [22]	✓	×	**Tamper Cls**	-
GSR-Net [45]	✓	×	ImageNet	-
CAT-Net [28]	✓	✓	ImageNet	-
			Double JPEG Det	-
MVSS-Net [8]	✓	✓	ImageNet	-
SATFL-Net [46]	✓	×	ImageNet	-
CA-IFL [40]	✓	×	ImageNet	-
			Wavelet Transform	-
PSCC-Net [31]	✓	✓	ImageNet	-
TruFor [16]	✓	✓	ImageNet	-
HiFi-Net [17]	✓	✓	ImageNet	-
Ours	✓	✓	ADE20k	**DDPM**

Table 1. Weight allocation methods for different IFDL methods, ”-” indicates random initialization of weights, and bold represents the micro-weights designed for the IFDL task.

Image Forgery Detection and Localization. Most of the existing methods perform pixel-wise classification to identify forged regions [8, 16, 17, 30, 31, 45, 46] employ ImageNet pre-trained weights as the foundation for their feature extraction encoders in tamper detection tasks. These methods try to improve the detection performance of tampered images by exploring the macroscopic features. However, they may suffer from degradation in terms of generality and robustness when dealing with unseen tampered images or unknown attacks. Recent methods [5, 7, 18, 21, 22, 28, 40, 42] aim to discover more effective tampering micro-features through self-supervised learning, with the goal of enhancing IFDL performance. Mantra-Net [42] and SPAN [22] design a self-supervised learning task to learn robust image manipulation traces. CAT-Net [28] performs double compression detection on JPEG images to obtain an encoder with microscopic feature weights and a parallel combination of macroscopic feature weights to form a dual-stream network

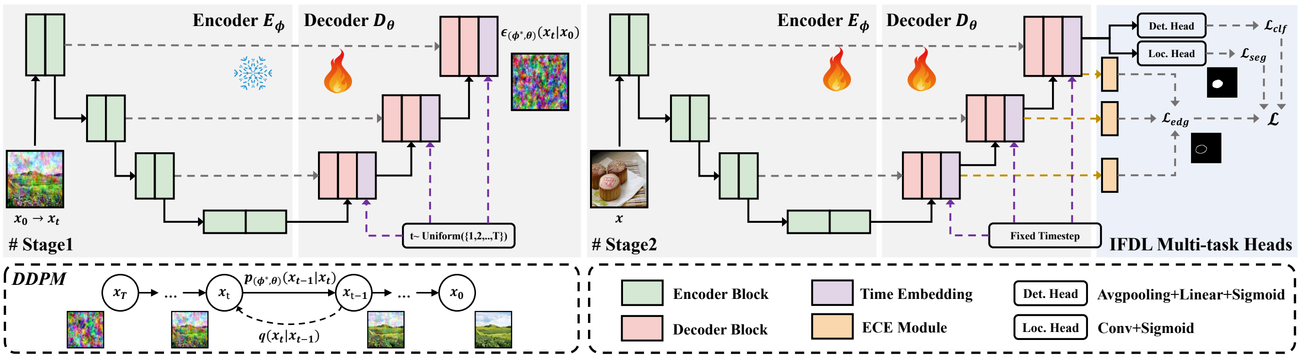


Figure 2. The overall framework of the proposed DiffForensics. The training process consists of two stages, *i.e.*, Stage 1: Self-supervised denoising diffusion pretraining (left), and Stage 2: Multi-task fine-tuning (right).

to improve the splicing detection performance of JPEG images. CA-IFL [40] and Bi *et al.* [5] respectively propose a wavelet-based representation learning strategy and design a JPEG compression operation chain tracker for pre-training to obtain microscopic feature weights with the ability to learn JPEG compression traces, which are used to improve the localization performance against JPEG compression. Chen *et al.* [7] and Hu *et al.* [21] reconstruct real or tampered faces through masks, and RealForensics [18] compares the dense connections between different modalities. These methods [7, 18, 21] seek to learn the micro features with better representation capabilities and improve generalization performance in the face of cross-dataset testing. According to Table 1, however, the training strategy by retaining either macro-feature or micro-feature weights in the encoder while randomly initializing the decoder weights, could by no means take full advantage of these two types of features in IFDL tasks.

In this paper, we propose a novel training scheme for the encoder-decoder model. For the encoder, we utilize pre-trained weights from a semantic segmentation task and freeze them to extract comprehensive macroscopic features. For the decoder, we introduce a DDPM-based paradigm to capture intricate microscopic features. Incorporating the above process steers the model to focus on the mesoscopic properties of images. Such a concentration is advantageous for the subsequent fine-tuning stage, enabling the model more precisely for the IFDL task.

3. The Proposed Method

In this section, we begin by presenting an overview of DiffForensics, which is illustrated in Fig. 2. As for the architecture, our approach comprises an encoder E_ϕ and a decoder D_θ which are parameterized by two sets of weights ϕ and θ, respectively. The training progress of our proposed framework contains two stages: self-supervised denoising diffusion pretraining and multi-task fine-tuning. Subsequent subsections will provide the details of each stage.

Algorithm 1 Denoising Diffusion Pre-training

```
repeat
    $x_0 \sim q\left(x_0\right)$
    $t \sim \text{Uniform}(\{1, 2, \ldots, T-1, T\})$
    Randomly generate simplex seed
    $\epsilon \sim \text{Simplex}(\nu = 2^{-6}, N = 6, \gamma = 0.8)$
    Take gradient descent step on
        $\nabla_{(\phi^*,\theta)} \left[ \left\| \epsilon - \epsilon_{(\phi^*,\theta)}(x_0\sqrt{\bar{a}_t} + \sqrt{1-\bar{a}_t}\epsilon, t) \right\|^2 \right]$
until converged
```

3.1. Self-supervised Denoising Diffusion Pre-training

Pipeline. In this stage, we aim to make the model focus on the *mesoscopic* proprieties of images, which can be further effectively fine-tuned for the IFDL task.

For the encoder, we utilize the transformer encoder blocks from SegFormer [44], and apply the pre-trained weights ϕ^* from a semantic segmentation task (*e.g.*, ADE20K) . We freeze the weights to retain the ability to extract *macroscopic* semantic features. For the decoder, we employ decoder blocks commonly used in Unet [37]. Consider that DDPM [19] consists of two opposite processes adding noise and reverse denoising, it can effectively learn the *microscopic* noise representations of the image. Motivated by this, we propose a denoising diffusion-based paradigm as the self-supervised pretext task to optimize the θ, without utilizing the forgery supervision. The overall training process is shown in the left part of Fig. 2 and detailed in Algorithm 1.

Specifically, given an image $x_0 \in \mathbb{R}^{3\times h\times w}$, and the time step t, we corrupt x_0 by adding noise ϵ via the diffusion process $q(x_t|x_{t-1})$, and perform the inverse process $p_{(\phi^*,\theta)}(x_{t-1}|x_t)$ to estimate the noise as $\epsilon_{(\phi^*,\theta)}(x_t|x_0) = D_\theta(E_\phi(x_0), t)$, and then to denoise. In this manner, we

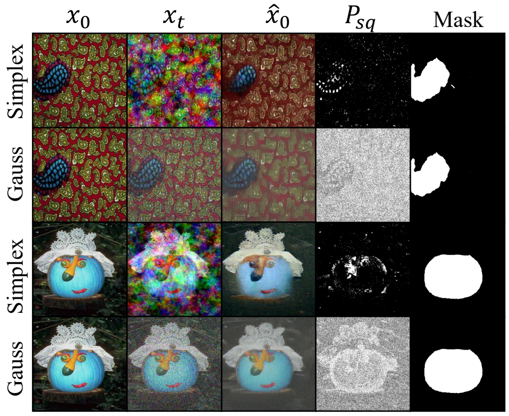


Figure 3. Qualitative denoising diffusion results of Simplex noise and Gaussian noise. From the square error prediction P_{sq} ($P_{sq} = (x_0 - \hat{x}_0)^2$), it can be seen that Simplex noise is more able to perceive tampered areas, while Gaussian noise simply restores the overall situation.

train the whole autoencoder model $E^*_\phi \circ D_\theta$ (*i.e.*, frozen encoder and the trainable decoder) to minimize the reconstruction error objective function as follows:

$$\ell_s = \mathbb{E}_{t\in[1,T],x_0\sim q(x_0),\epsilon\sim\mathcal{S}(\nu,N,\gamma)}[||\epsilon - \epsilon(\phi^*,\theta)||^2]. \quad (1)$$

By combining the above *macroscopic* and *microscopic* representations, we guide the whole auto-encoder $E^*_\phi \circ D_\theta$ to concentrate on *mesoscopic* features of images.

Simplex noise. Different from the vanilla DDPM [19], we destroy x_0 by adding Simplex noise [43] instead of the Gaussian noise in the diffusion process. As shown in Fig. 3, the potential benefit of such noise over the standard Gaussian perturbations is intuitive: the corruption of images is more structured (*e.g.*, the edge of tampered regions) and the denoising process will be able to "repair" them, thereby facilitating the learning of such structured anomalies. For the hyper-parameters of Simplex noise $\epsilon \sim \mathcal{S}(\nu, N, \gamma)$, we set a starting frequency $\nu = 2^{-6}$, octave $N = 6$ and a decay $\gamma = 0.8$

3.2. Multi-task Fine-Tuning

Pipeline. After pretraining, we fine-tune the pre-trained autoencoder (both encoder and decoder) on data with the IFDL supervision (*i.e.*, the forgery label and mask). According to our ablation studies, multi-task learning can help learn better representative features with good performance. Therefore, we add multi-task heads (*i.e.*, the detection and localization heads) in the latter of the decoder, as depicted in the right part of Fig. 2.

Edge Cue Enhancement Module. To further mine the subtle traces of tampered regions, we introduce an Edge Cue Enhancement Module to enhance the edge cues on the output features of three scales decoder blocks in both horizontal and vertical directions, as illustrated in Fig. 4.

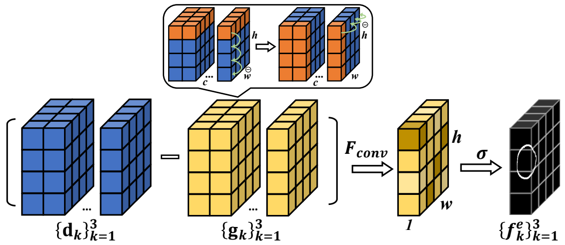


Figure 4. Details of Edge Cue Enhancement Module. Enhanced edge cues are employed in two dimensions to locate the boundaries of the tampered regions.

Specifically, let $\{\mathbf{d}_k\}^3_{k=1}$ be the output feature maps of each decoder block. Note that $\mathbf{d}_k \in \mathbb{R}^{b\times c\times h\times w}$ is a four-dimensional feature vector, we only conduct the following process in last two dimensions (*i.e.*, the height an the width) of $\mathbf{d}_k$. Initially, we compute the difference between adjacent rows in $\mathbf{d}_k$, and then take the absolute value to maintain consistent gradient orientation. This absolute difference is reassigned to the current row, enhancing the edge cue feature map in the row direction. Subsequently, we apply the same process to the columns of the enhanced features, where the difference between adjacent columns is calculated and its absolute value is taken to ensure gradient orientation consistency. In this manner, we obtain the edge-enhanced features of $\mathbf{d}_k$, denoted as $\mathbf{g}_k$. The above pipeline can be formulated as:

$$\mathbf{g}_k = |\mathbf{V} * |\mathbf{H} * \mathbf{d}_k|| \quad (2)$$

where $*$ is the convolution operation, and $|\cdot|$ is the abs operation. $\mathbf{H} = [1, -1]$ and $\mathbf{V} = [1, -1]^\top$ are the edge enhancement operators in the horizontal and vertical direction, respectively.

After that, we compute the difference between $\mathbf{d}_k$ and $\mathbf{g}_k$ and employ a 3×3 convolution to reduce the dimension, and finally use the sigmoid function to normalize the cue feature map to 0-1, and finally up-sample to the same size as the input image to obtain our edge prediction probability map f^e_k, which can be marked as:

$$f^e_k = U\left(\sigma\left(F_{cov}\left(\mathbf{d}_k - \mathbf{g}_k\right)\right)\right). \quad (3)$$

where F_{cov} is a 3×3 convolution operation, σ is sigmoid normalization, U is an up-sampling operation, and the obtained edge prediction probability map f^e_k and edge label y^e of each decoder are used for loss iteration. We employ the above ECEM across all three scales of $\mathbf{d}_k$.

Loss function. There are three types of supervision in our method, *i.e.*, localization segmentation supervision $\mathcal{L}_{seg}$, detection classification supervision $\mathcal{L}_{clf}$, and edge cue supervision $\mathcal{L}_{edg}$.

For pixel-level localization segmentation supervision, we use a combination of weighted ℓ_{wbce} and ℓ_{dice} [32].

$$\mathcal{L}_{seg}\left(x\right) = \lambda_0^s \ell_{wbce} + \left(1 - \lambda_0^s\right) \ell_{dice}. \tag{4}$$

where λ_0^s is the segmentation balance weight, and the weighted segmentation ℓ_{wbce} and ℓ_{dice} are respectively:

$$\begin{aligned}\ell_{wbce} = -\frac{1}{N}\textstyle\sum_{i,j}\big(&\lambda_1^s \cdot y_{i,j}^s \cdot \log f^s\left(x_{i,j}\right) \\ &+ \lambda_2^s \cdot (1 - y_{i,j}^s) \cdot \log\left(1 - f^s\left(x_{i,j}\right)\right)\big).\end{aligned} \tag{5}$$

$$\ell_{dice} = 1 - \frac{2\sum_{i,j} f^s(x_{i,j}) \cdot y_{i,j}^s}{\sum_{i,j}(f^s(x_{i,j}))^2 + \sum_{i,j}(y_{i,j}^s)^2}. \tag{6}$$

where $y_{i,j}^s \in \{0, 1\}$ is a pixel-level binary label, representing whether the $\{i, j\}$ th pixel has been tampered with. λ_1^s and λ_2^s are the weights of balancing tampered pixels and real pixels, respectively, and encourage the network to pay more attention to those difficult pixel samples.

For edge supervision, we use the same dice loss as the above segmentation supervision, but here, to standardize the edge of tampering position step by step from coarse-grained to fine-grained, the probability map $\{f_k^e\}_{k=1}^3$, we designed multi-scale supervision weights, aiming to give fine-grained edge supervision greater weight, while standardizing coarse-grained edge supervision, so that f_k^e is better refined One-stage fine-grained edge supervision f_{k-1}^e.

$$\mathcal{L}_{edg}\left(x\right) = \sum_{k=1}^{3} \frac{1}{2^{k-1}} \ell_{dice}\left(f_k^e, y^e\right). \tag{7}$$

For image-level detection and classification supervision, in order to alleviate the imbalance of positive and negative samples of image-level data, we use weighted ℓ_{wbce}.

$$\begin{aligned}\mathcal{L}_{clf}(x) = -(&\lambda_0^c \cdot y^c \cdot \log f^c(x) \\ &+ \lambda_1^c \cdot (1 - y^c) \cdot \log(1 - f^c(x))).\end{aligned} \tag{8}$$

where y^c is the image-level binary label, and $f^c\left(x\right)$ is the classification prediction result. Since the number of positive and negative samples at the image level is easy to measure, we automatically set the tampering weight as $\lambda_0^c = \left\lfloor \frac{10 * Num_F}{Num_{F+R}} \right\rfloor /10$, and set the real weight as $\lambda_1^c = \left\lfloor \frac{10 * Num_R}{Num_{F+R}} \right\rfloor /10$, Num_F and Num_R represent the number of falsified images and real images respectively.

Finally, we define the total loss $\mathcal{L}$ as a weighted combination of above three losses, formulated as:

$$\mathcal{L} = \alpha \cdot \left(\mathcal{L}_{seg} + \mathcal{L}_{edg}\right) + \beta \cdot \mathcal{L}_{clf}. \tag{9}$$

where $\alpha, \beta \in [0, 1]$.

4. Experiments

4.1. Experimental Setup

Dataset. Considering the availability and generality, we select some challenging benchmark datasets to evaluate our method, among which CASIAv2.0 [14], Fantasitic-Reality [26], CASIAv1+ [8], Columbia [20], NIST16 [15], IMD2020 [34], DSO-1 [12] and Korus [27] are tampered by traditional image editing tools, while AutoSplicing [23] and OpenForensics [29] are tampered by deep generative models (DGMs). Details of these datasets are provided in the Appendix, and the configuration details at different stages are as follows:

(1) Denoising diffusion pretraining: We mixed all data (both forgery and authentic) of CASIAv2.0 [14] and Fantasitic-Reality [26] for the self-supervised pretraining, which do not use the forgery supervision in this stage.

(2) Multi-task fine-tuning: We also utilized the CASIAv2.0 [14] and Fantasitic-Reality [26] datasets with their forgery supervision. Note that we only use the forgery images for the Fantasitic-Realiy [26] dataset for the balance of the number of the forgery and authentic pixels overall.

(3) Evaluation: To verify the generalization performance, we evaluated our method on other image editing forgery datasets, *i.e.*, CASIAv1+ [8], Columbia [20], NIST16 [15], IMD2020 [34], DSO-1 [12] and Korus [27] datasets. We also utilized two recent datasets forged by the advanced DGMs, *i.e.*, AutoSplicing [23] and OpenForensics [29].

Implementation details. We use 4 NVIDIA Tesla A100 GPUs (80 GB memory) to conduct experiments on the PyTorch deep learning framework. We perform the following parameter configurations for the two stages:

(1) Denoising diffusion pretraining: In the pre-training stage, we resized the input image to 512×512 and applied the AdamW optimizer. We set the training hyper-parameters by the learning rate as 10^{-4}, the diffusion steps T as 1000, the batch size as 16, and the epoch as 100.

(2) Multi-task fine-tuning: In the fine-tuning stage, we also resized the input image to 512×512 and applied the AdamW optimizer. We set the training hyper-parameters by the learning rate as 10^{-4}, the batch size as 32, and the epoch as 50, the fixed time embedding as $t = 5$ (Details can be seen in ablation study). To balance the performance of forgery detection and localization, we set the weight of tamper localization $\mathcal{L}_{seg}$ and edge supervision $\mathcal{L}_{edg}$ to $\alpha = 0.8$, where λ_0^s, λ_1^s and λ_2^s in $\mathcal{L}_{seg}$ are 0.1, 2 and 0.5, respectively. The weight β of the supervision $\mathcal{L}_{clf}$ for tamper detection is set to 0.1, and λ_0^c and λ_1^c are 0.7 and 0.3, respectively.

Evaluation metrics. For forgery localization, we report pixel-level F1 and AUC (Area Under Curve of a Receiver-Operating-Characteristic curve). For forgery detection, in addition to image-level ACC and AUC, we further report the EER (Equal Error Rate) to evaluate the false alarm and

Methods	**Editing**												**DGM**				**Average**	
	CASIA1.0+		Columbia		NIST16		IMD2020		DSO-1		Korus		AutoSplice		OpenForensics			
	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC
H-LSTM [3]	.121	.532	.257	.558	.109	.573	.118	.570	.187	.559	.089	.536	.306	.598	.123	.622	.164	.569
ManTra-Net* [42]	.136	.612	.357	.767	.160	.741	.180	.785	.089	.687	.104	.681	.192	.622	.043	.678	.158	.697
HP-FCN [30]	.132	.772	.050	.549	.071	.690	.029	.665	.013	.545	.076	.663	.029	.556	.027	.650	.053	.636
GSR-Net [45]	.244	.821	.340	.836	.221	.746	.102	.788	.055	.697	.060	.632	.047	.722	.025	.683	.137	.741
SPAN [22]	.088	.533	.213	.597	.116	.648	.108	.671	.059	.564	.070	.575	.047	.572	.014	.682	.089	.605
MVSS-Net* [8]	.451	.845	.665	.818	.292	.791	.264	.817	.271	.732	.095	.641	.333	.839	.056	.702	.303	.773
CAT-Net [28]	.394	.788	.854	.826	.336	.780	.295	.823	.135	.713	.149	.672	.185	.796	.003	.552	.294	.744
SATL-Net [46]	.064	.545	.677	.872	.175	.655	.142	.671	.084	.575	.039	.577	.103	.590	.019	.544	.163	.629
PSCC-Net [31]	.355	.738	.672	.881	.238	.740	.295	.800	.318	.721	.156	.623	.150	.784	.065	.610	.281	.737
HiFi-Net [17]	.092	.642	.382	.608	.172	.685	.178	.675	.304	.700	.088	.607	**.613**	.831	**.149**	.676	.247	.678
Ours	**.517**	**.868**	**.912**	**.931**	**.415**	**.828**	**.511**	**.911**	**.485**	**.874**	**.257**	**.721**	.507	**.940**	.122	**.820**	**.466**	**.862**

Table 2. Pixel-level F1 and AUC performance of image forgery localization. The best result is highlighted and bold. Except the method with ∗ uses the pre-training model of the original paper, other methods keep the same training data as our method.

missed detection performances. For both forgery detection and localization, the default threshold is 0.5 unless otherwise specified.

4.2. Comparison with the State-of-the-Art Methods

For a fair comparison, we focus on methods with available codes or pre-trained models as follows.

(1) Pre-trained models available: To avoid biases, we only included the methods trained on datasets different from the test datasets. ManTra-Net [42] is pre-trained on a million private dataset. MVSS-Net [8] is pre-trained on the CASIA2 dataset. For these methods, we directly use their pre-trained models for evaluation.

(2) Code available: H-LSTM [3], HP-FCN [30], GSR-Net [45], SPAN [22], SATL-Net [46], CAT-Net [28], PSCC-Net [31] and HiFi-Net [17]. For these methods, we retrained them with the same experimental settings as ours, and using the optimal hyper-parameter configurations.

Localization evaluation. Table 2 shows the forgery localization performance. We observed that our method achieves superior performances on all datasets. It is worth mentioning that HiFi-Net, which is specially designed for DGM forgery detection and localization, achieved the best F1 score on DGM forgery datasets. In general, our proposed method achieves the best average performance, which demonstrates its effectiveness.

Detection evaluation. Following [8, 31], we conducted the evaluation of image-level classification using datasets with both authentic and tampered images. Table 3 shows the forgery detection performance. We observed that our method also achieves superior performances on all datasets. In general, our proposed method achieves the best average AUC, EER and the second-best ACC, which also demonstrates its effectiveness. It should be noted that, for datasets with extremely imbalanced positive and negative samples, *e.g.*, IMD2020 [34] (authentic: 414, tampered: 2010), the metric relevant to threshold could not evaluate the overall performance. Although our method does not show a better ACC score for threshold 0.5, it achieves better overall performance in terms of AUC score, and a superior balanced error rate in terms of EER.

Robustness. We further evaluated the robustness when facing common image perturbations in social media laundering, *i.e.*, JPEG compression and Gaussian noising. We reported the average of F1 and AUC scores as the indicator. It can be seen that our method shows better robust performance in both forgery localization and forgery detection tasks. Especially in the forgery localization, with the dual support of macro-features and micro-features, it has achieved a substantial performance lead.

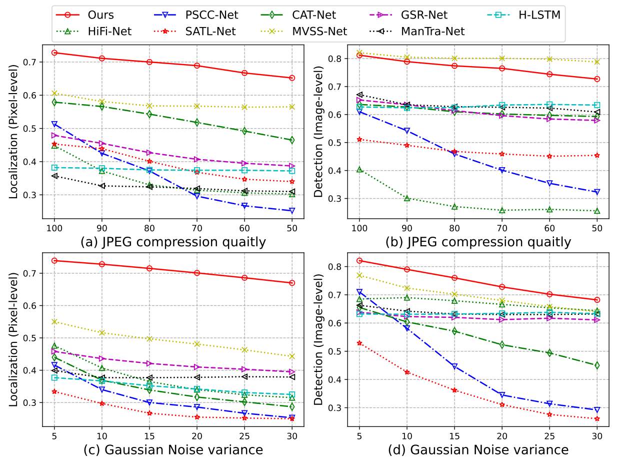
4.3. Ablation Study

This section analyzes the effectiveness of several key components in the proposed two-stage training stages.

Self-supervised denoising diffusion pre-training. In this part, we analyzed the impacts of *diffuse noise* and *model weights* in the denoising diffusion pre-training. As shown in Table 4, we verified the performance of the choice of diffuse noise under different weight combinations. First, the 1st row does not perform DDPM pre-training baseline, the 2nd and 3rd rows use Gaussian noise for DDPM pre-training, and the 4th and 5th rows use Simplex noise for DDPM pre-training. Comparing the 2nd and 3rd rows, and comparing the 4th and 5th rows, it can be seen that pre-training using Simplex noise achieves better results on both artificially tampered and synthetically tampered datasets, which shows that the Simplex noise has a greater impact on micro tampering. The perceptual learning of traces is more pronounced. The weight of loading is also the focus of this article. By comparing rows 1, 3, and 5, we can see that the combination strategy of encoder macro feature extraction and decoder micro feature extraction proposed in this paper can effectively improve the performance of IFDL tasks. By comparing rows 2 and 4 with the other three rows, it can

Methods	Editing									DGM			Average		
	CASIA1.0+			Columbia			IMD2020			AutoSplice					
	ACC↑	AUC↑	EER↓	ACC↑	AUC↑	EER↓	ACC↑	AUC↑	EER↓	ACC↑	AUC↑	EER↓	ACC↑	AUC↑	EER↓
H-LSTM [3]	.535	.490	.550	.496	.506	.443	.829	.494	.556	.614	.500	.542	.619	.498	.523
ManTra-Net [42]	.535	.546	.446	.496	.869	.219	**.830**	.698	.372	.614	.378	.586	.619	.623	.406
GSR-Net [45]	.595	.657	.401	.540	.721	.333	.671	.511	.493	.568	.540	.469	.594	.607	.424
MVSS-Net [8]	.791	.937	.136	.664	**.984**	**.055**	.799	.661	.391	**.809**	.886	.191	.766	.867	.193
CAT-Net [28]	.671	.690	.362	.755	.953	.115	.785	.684	.370	.699	.790	.296	.728	.779	.286
SATL-Net [46]	.459	.392	.573	.744	.912	.131	.667	.602	.420	.463	.347	.614	.583	.563	.435
PSCC-Net [31]	**.992**	**.999**	**.006**	.606	.981	.082	.821	.624	.425	.733	.877	.192	**.788**	.870	.176
HiFi-Net [17]	.632	.717	.320	.532	.741	.317	.826	.523	.483	.618	.527	.457	.652	.627	.394
Ours	.741	.991	.043	**.895**	.982	**.055**	.749	**.740**	**.333**	.696	**.951**	**.092**	.770	**.916**	**.131**

Table 3. Image-level ACC, AUC and EER performance of image forgery detection. In the average test results on artificial editing data and deep synthetic data, our method obtains the highest AUC and the lowest EER performance, and the suboptimal ACC.



(a) JPEG compression quaitly

(b) JPEG compression quaitly

(c) Gaussian Noise variance

(d) Gaussian Noise variance

Figure 5. Robustness against jpeg compression and Gaussian noise effects. Tested on CASIA1.0+, Columbia, IMD2020 and AutoSplicing. Our method achieves a substantial lead in tamper localization performance.

be seen that the encoder's DDPM training may cause catastrophic forgetting of the original macroscopic features.

Furthermore, we show the embedding space of learned features with t-SNE [39] visualization in Fig. 7. We can observe that the combination of noise selection and encoder-decoder weight selection in the final scheme can effectively distinguish the feature distribution of real samples from tampered samples. The comprehensive results show that the training method proposed in this paper combines the macroscopic features with supervised weights and the microscopic features obtained by DDPM pre-training with Simplex noise to achieve the best IFDL performance.

Multi-task fine-tuning. Herein, we analyzed the impacts of the loss functions and time embedding t_f.

(1) Combination of loss functions: For $\mathcal{L}_{seg}$ and $\mathcal{L}_{clf}$, ℓ_{s1} and ℓ_{c1} represent weighted ℓ_{bce}, ℓ_{s2} and ℓ_{c2} represent unweighted ℓ_{bce}. For $\mathcal{L}_{edg}$ each parameter of (i) ℓ_{e1}: Add edge supervision with ECEM to the last decoder output, with a weight of 1. (ii) ℓ_{e2}: Add edge supervision with ECEM to all decoder outputs, but the weights are all 1. (iii) ℓ_{e3}: The multi-scale weighted edge supervision with ECEM proposed in this paper sets a smaller weight for the coarser-grained edge supervision, and sets a larger weight for the finer-grained edge supervision. By comparing the 1st row and the last row of Table 5, it can be seen that the multi-weight and multi-scale edge cue enhanced supervision loss not only greatly improves the tamper localization task, but also promotes the performance of the tamper detection task. By comparing the 2nd, 3rd, and last rows, it shows that this paper designs different weighting strategies for scale edges of different granularities, which can better enhance the traces of tampered areas of different scales. Finally, by comparing the 4th row, the 5th row, and the last row, weighting ℓ_{seg} and ℓ_{clf} respectively can achieve a certain performance improvement in IFDL.

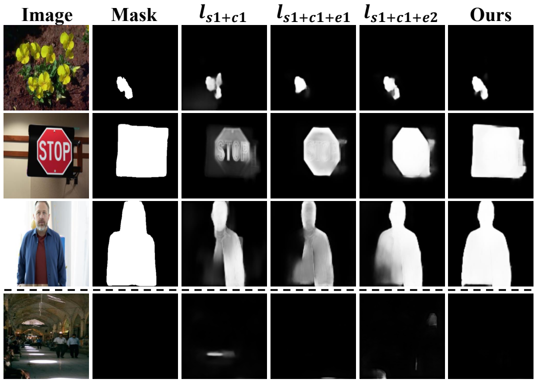


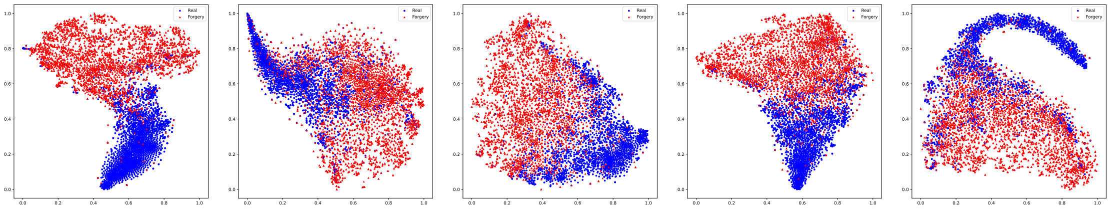
Figure 6. Visualizing the performance impact of ECEM in varied setups. The test image in the last row is authentic.

We also depict some qualitative results in Figure 6. From left to right, it is observed that the location and contour of the tampered region are more precisely localized under the supervision of the multi-scale edge cue enhancement module. Meanwhile, our method can also effectively lower the false alarm risk for authentic images.

(2) Fixed time embedding t_f: we use T ∈ [0,1000] for de-

Noise		**Model weights**		**Localization**				**Detection**				**Average**	
				IMD2020		AutoSplicing		IMD2020		AutoSplicing			
Gauss	Simplex	Encoder	Decoder	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC
-	-	ADE20K	-	.416	.904	.297	.910	.768	.725	.634	.862	.529	.850
✓	-	DDPM	DDPM	.372	.865	.340	.895	.737	.646	.573	.942	.506	.837
✓	-	ADE20K	DDPM	.470	.904	.346	.918	.738	.712	.604	.910	.540	.861
-	✓	DDPM	DDPM	.380	.855	.304	.881	.704	.660	.358	.835	.437	.808
-	✓	ADE20K	DDPM	**.511**	**.911**	**.507**	**.940**	**.841**	**.740**	**.679**	**.951**	**.635**	**.886**

Table 4. For IFDL tasks, the performance of different weight settings for DDPM pre-training using Simplex noise and Gaussian noise for encoder and decoder structures.



(a) ADE20K-Simplex (b) Simplex-Simplex (c) ADE20K-Gauss (d) Gauss-Gauss (e) ADE20K-None

Figure 7. Feature space visualization of different diffused noise selection and loaded weights. Tested on AutoSplicing.

$\mathcal{L}$			**Localization**		**Detection**		
$\mathcal{L}_{seg}$	$\mathcal{L}_{clf}$	$\mathcal{L}_{edg}$	F1	AUC	ACC	F1	AUC
ℓ_{s1}	ℓ_{c1}	-	.316	.867	.686	.730	.783
ℓ_{s1}	ℓ_{c1}	ℓ_{e1}	.409	.894	.659	.679	.813
ℓ_{s1}	ℓ_{c1}	ℓ_{e2}	.412	.910	.623	.635	.808
ℓ_{s1}	ℓ_{c2}	ℓ_{e3}	.452	.910	.680	.728	.813
ℓ_{s2}	ℓ_{c1}	ℓ_{e3}	.387	.899	.660	.696	.843
ℓ_{s1}	ℓ_{c1}	ℓ_{e3}	**.509**	**.925**	**.722**	**.760**	**.846**

Table 5. The IFDL performance of the combination of three levels of supervision loss and the ablation performance of the ECEM is represented in $\mathcal{L}_{edg}$. Tested on IMD2020 and AutoSplicing.

t_f	**Localization**		**Detection**		**Average**	
	F1	AUC	F1	AUC	F1	AUC
0	.385	.885	.640	.840	.513	.863
1	.427	.920	**.783**	.847	.605	.884
3	**.515**	.914	.692	**.856**	.604	.885
5	.509	**.925**	.760	.846	**.635**	**.886**
10	.366	.894	.686	.801	.526	.848
25	.388	.894	.707	.742	.548	.818
125	.384	.882	.635	.799	.510	.841
250	.437	.901	.695	.812	.566	.857
500	.312	.904	.577	.809	.445	.857
750	.507	.883	.570	.847	.539	.865
1000	.389	.887	.707	**.856**	.548	.872

Table 6. Ablation of the t_f in multi-task fine-tuning stage. Tested on IMD2020 and AutoSplicing.

noising diffusion pre-training and adopt fixed timestep t_f for training and testing during multi-task fine-tuning. To optimize the t_f for better feature representation, we conduct the grid search at t $\in$ [0,1000], and the results are summarized in Table 6. It is observed that the smaller t is beneficial to learn the tampering traces, as a result, we use t_f = 5 for the time embedding parameter.

5. Conclusion

In this study, we propose a novel two-stage self-supervised method with an encoder-decoder structure for the image forgery detection and localization task. At the first denoising diffusion pre-training stage, the encoder pre-trained on the segmentation task is frozen while the decoder is trained with a self-supervised denoising diffusion paradigm. It aims to encourage the model to concentrate on the *mesoscopic* proprieties of images. After pre-training, we fine-tune the pre-trained model with a supervised multi-task framework and introduce an edge cue enhancement module in the decoder to enhance tampering traces from coarse to fine. Extensive experimental results demonstrate that our proposed method achieves superior performances compared to state-of-the-art competitors on several emerging datasets (including artificially manipulated and AI-generated images) in terms of detection and localization performances.

Acknowledgements This work was partially supported by the National Natural Science Foundation of China (Grants No. U23B2022, U22A2030, U22B2047), Guangdong Major Project of Basic and Applied Basic Research (Grand No. 2023B0303000010) and Special Fund for the Cultivation of Science and Technology Innovation for Guangdong University Students (pdjh2022b0444).

References

[1] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. Mesonet: a compact facial video forgery detection network. In *2018 IEEE international workshop on information forensics and security (WIFS)*, pages 1–7. IEEE, 2018. 2

[2] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18208–18218, 2022. 1, 2

[3] Jawadul H Bappy, Cody Simons, Lakshmanan Nataraj, BS Manjunath, and Amit K Roy-Chowdhury. Hybrid lstm and encoder–decoder architecture for detection of image forgeries. *IEEE Transactions on Image Processing*, 28(7):3286–3300, 2019. 2, 6, 7

[4] Dmitry Baranchuk, Ivan Rubachev, Andrey Voynov, Valentin Khrulkov, and Artem Babenko. Label-efficient semantic segmentation with diffusion models. *arXiv preprint arXiv:2112.03126*, 2021. 2

[5] Xiuli Bi, Wuqing Yan, Bo Liu, Bin Xiao, Weisheng Li, and Xinbo Gao. Self-supervised image local forgery detection by jpeg compression trace. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 232–240, 2023. 2, 3

[6] Emmanuel Asiedu Brempong, Simon Kornblith, Ting Chen, Niki Parmar, Matthias Minderer, and Mohammad Norouzi. Denoising pretraining for semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4175–4186, 2022. 2

[7] Han Chen, Yuzhen Lin, Bin Li, and Shunquan Tan. Learning features of intra-consistency and inter-diversity: Keys toward generalizable deepfake detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(3):1468–1480, 2022. 2, 3

[8] Xinru Chen, Chengbo Dong, Jiaqi Ji, Juan Cao, and Xirong Li. Image manipulation detection by multi-view multi-scale supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14185–14193, 2021. 1, 2, 5, 6, 7

[9] Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. Ilvr: Conditioning method for denoising diffusion probabilistic models. *arXiv preprint arXiv:2108.02938*, 2021. 2

[10] Hyungjin Chung, Byeongsu Sim, and Jong Chul Ye. Come-closer-diffuse-faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12413–12422, 2022. 2

[11] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 2

[12] Tiago José De Carvalho, Christian Riess, Elli Angelopoulou, Helio Pedrini, and Anderson de Rezende Rocha. Exposing digital image forgeries by illumination color classification. *IEEE Transactions on Information Forensics and Security*, 8 (7):1182–1194, 2013. 5

[13] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 2

[14] Jing Dong, Wei Wang, and Tieniu Tan. Casia image tampering detection evaluation database. In *2013 IEEE China summit and international conference on signal and information processing*, pages 422–426. IEEE, 2013. 5

[15] Haiying Guan, Mark Kozak, Eric Robertson, Yooyoung Lee, Amy N Yates, Andrew Delgado, Daniel Zhou, Timothee Kheyrkhah, Jeff Smith, and Jonathan Fiscus. Mfc datasets: Large-scale benchmark datasets for media forensic challenge evaluation. In *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pages 63–72. IEEE, 2019. 5

[16] Fabrizio Guillaro, Davide Cozzolino, Avneesh Sud, Nicholas Dufour, and Luisa Verdoliva. Trufor: Leveraging all-round clues for trustworthy image forgery detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20606–20615, 2023. 1, 2

[17] Xiao Guo, Xiaohong Liu, Zhiyuan Ren, Steven Grosz, Iacopo Masi, and Xiaoming Liu. Hierarchical fine-grained image forgery detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3155–3165, 2023. 1, 2, 6, 7

[18] Alexandros Haliassos, Rodrigo Mira, Stavros Petridis, and Maja Pantic. Leveraging real talking faces via self-supervision for robust forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14950–14962, 2022. 2, 3

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2, 3, 4

[20] Yu-Feng Hsu and Shih-Fu Chang. Detecting image splicing using geometry invariants and camera characteristics consistency. In *2006 IEEE International Conference on Multimedia and Expo*, pages 549–552. IEEE, 2006. 5

[21] Juan Hu, Xin Liao, Difei Gao, Satoshi Tsutsui, Qian Wang, Zheng Qin, and Mike Zheng Shou. Mover: Mask and recovery based facial part consistency aware method for deepfake video detection. *arXiv preprint arXiv:2305.05943*, 2023. 2, 3

[22] Xuefeng Hu, Zhihan Zhang, Zhenye Jiang, Syomantak Chaudhuri, Zhenheng Yang, and Ram Nevatia. Span: Spatial pyramid attention network for image manipulation localization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*, pages 312–328. Springer, 2020. 1, 2, 6

[23] 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. 5

[24] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 1

[25] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020. 1

[26] Vladimir V Kniaz, Vladimir Knyaz, and Fabio Remondino. The point where reality meets fantasy: Mixed adversarial generators for image splice detection. *Advances in neural information processing systems*, 32, 2019. 5

[27] Paweł Korus and Jiwu Huang. Multi-scale analysis strategies in prnu-based tampering localization. *IEEE Transactions on Information Forensics and Security*, 12(4):809–824, 2016. 5

[28] Myung-Joon Kwon, In-Jae Yu, Seung-Hun Nam, and Heung-Kyu Lee. Cat-net: Compression artifact tracing network for detection and localization of image splicing. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 375–384, 2021. 2, 6, 7

[29] Trung-Nghia Le, Huy H Nguyen, Junichi Yamagishi, and Isao Echizen. Openforensics: Large-scale challenging dataset for multi-face forgery detection and segmentation in-the-wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10117–10127, 2021. 5

[30] Haodong Li and Jiwu Huang. Localization of deep inpainting using high-pass fully convolutional network. In *proceedings of the IEEE/CVF international conference on computer vision*, pages 8301–8310, 2019. 1, 2, 6

[31] 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. 1, 2, 6, 7

[32] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. Ieee, 2016. 5

[33] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021. 2

[34] 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. 5, 6

[35] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 2

[36] 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. 1, 2

[37] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 3

[38] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2

[39] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9 (11), 2008. 7

[40] Menglu Wang, Xueyang Fu, Jiawei Liu, and Zheng-Jun Zha. Jpeg compression-aware image forgery localization. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 5871–5879, 2022. 2, 3

[41] Julia Wolleb, Florentin Bieder, Robin Sandkühler, and Philippe C Cattin. Diffusion models for medical anomaly detection. In *International Conference on Medical image computing and computer-assisted intervention*, pages 35–45. Springer, 2022. 2

[42] 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. 1, 2, 6, 7

[43] Julian Wyatt, Adam Leach, Sebastian M Schmon, and Chris G Willcocks. Anoddpm: Anomaly detection with denoising diffusion probabilistic models using simplex noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 650–656, 2022. 2, 4

[44] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021. 2, 3

[45] Peng Zhou, Bor-Chun Chen, Xintong Han, Mahyar Najibi, Abhinav Shrivastava, Ser-Nam Lim, and Larry Davis. Generate, segment, and refine: Towards generic manipulation segmentation. In *Proceedings of the AAAI conference on artificial intelligence*, pages 13058–13065, 2020. 1, 2, 6, 7

[46] Long Zhuo, Shunquan Tan, Bin Li, and Jiwu Huang. Self-adversarial training incorporating forgery attention for image forgery localization. *IEEE Transactions on Information Forensics and Security*, 17:819–834, 2022. 1, 2, 6, 7