Fall 2025

Improving Generalizability in Image Manipulation Detection

Zhenfei Zhang
University at Albany, State University of New York, zzhang45@albany.edu

IMPROVING GENERALIZABILITY IN IMAGE MANIPULATION DETECTION

by

Zhenfei Zhang

A Dissertation

Submitted to the University at Albany, State University of New York

in Partial Fulfillment of

the Requirements for the Degree of

Doctor of Philosophy

College of Nanotechnology, Science, and Engineering

Department of Computer Science

Fall 2025

To my family for their unconditional love and support!

ABSTRACT

Image manipulation detection (IMD) aims to determine whether an image has been tampered with and to identify the manipulated regions. These capabilities have become increasingly important with the rapid advancement of media editing and generation technologies, such as Photoshop and generative AI methods, which underscore the need for robust tools for media authentication. Although current state-of-the-art (SoTA) methods achieve strong results on common manipulation types, such as splicing, copy-move, and removal, they often struggle to generalize to manipulation types not represented in the training data. Consequently, their real-world applicability remains limited, with performance degrading significantly in practical scenarios.

In this dissertation, we advance image manipulation detection through three key contributions. First, we introduce the *Challenging Image Manipulation Detection (CIMD) benchmark*, a novel, high-quality dataset with fine-grained annotations designed to evaluate SoTA methods on both editing-based and compression-based manipulations under realistic and more complex conditions. Using CIMD, we demonstrate that existing SoTA methods struggle with small tampered regions and double-compression cases with identical quality factors, and we propose a two-branch HRNet-based model that significantly outperforms prior work on these challenges. Second, we present a *unified unsupervised and weakly supervised framework* that reduces reliance on pixel-level annotations. This framework leverages implicit neural representations and selective contrastive learning, achieving detection performance comparable to supervised methods while improving robustness to unseen manipulations. Finally, we develop a *training-free diffusion-based approach* that exploits inconsistencies between conditional and unconditional reconstructions for manipulation detection. This method requires no external training data and outperforms existing unsupervised and weakly supervised techniques, while achieving competitive results with fully supervised models across multiple benchmark datasets.

Collectively, these contributions strengthen IMD performance in realistic tampering scenar-

ios and broaden its applicability to forensic settings where manipulation types are diverse, rapidly evolving, and often unseen during training.

ACKNOWLEDGMENT

I would first like to express my deepest gratitude to my PhD supervisor, Dr. Ming-Ching Chang, whose professionalism, guidance, and patience have profoundly shaped my doctoral journey. I am truly grateful for the opportunities you have provided and for your invaluable mentorship throughout this process.

I am also sincerely thankful to my dissertation committee members, Dr. Xin Li, Dr. Siwei Lyu, and Dr. Petko Bogdanov, for their insightful feedback and thoughtful advice, which have greatly strengthened my thesis.

I would like to extend my appreciation to my labmates, An Yu, Yuwei Chen, Abhineet Pandey, and Ting-Yu Tsai, for their help, collaboration, and encouragement along the way.

Finally, I am deeply indebted to my family. To my parents, for their unwavering love and support; to my grandparents, for their constant encouragement; and to my fiancée, for her invaluable companionship and love. I could not have completed this PhD without their support.

CONTENTS

LIST OF FIGURES

CHAPTER 1

Introduction

Digital images play a central role in communication and everyday life. People naturally trust what they see, believing their eyes will not deceive them. However, with the rapid development of AI-driven image generation and editing technologies, along with commercial tools such as Photoshop, manipulating images has become easier and more accessible than ever. Modern editing methods can produce highly realistic results that are extremely difficult for humans to distinguish from authentic content. As a consequence, manipulated images are no longer limited to entertainment or harmless creativity—they now pose serious risks, including the spread of misinformation, fraud, and threats to the safe deployment of AI systems. These concerns make it essential to develop robust image manipulation detection tools.

Image manipulation detection aims to determine whether an image has been altered and, if so, to localize the manipulated regions. This task has gained increasing attention in recent years. Despite encouraging progress in existing work, most prior approaches target only a narrow subset of manipulation types (e.g., splicing, copy–move, and removal) and depend on specific assumptions regarding image characteristics. In real-world scenarios, however, manipulations are diverse, unconstrained, and frequently generated by modern AI models. As a result, many existing methods struggle to generalize beyond the limited conditions for which they were designed.

This lack of generalizability motivates the work presented in this thesis. The central goal is to improve the robustness and adaptability of image manipulation detection methods so that they remain effective under diverse and realistic real-world scenarios. This thesis contributes to this goal in three ways. First, we introduce a novel and more challenging dataset, along with a new method capable of handling more complex real-world cases without requiring expanded training data. Second, we develop a unified weakly supervised and unsupervised framework that requires only minimal image-level supervision—or no labels at all—while remaining effective on complex

and unseen manipulation types. Third, we propose a training-free method that does not rely on any training data and can generalize naturally to previously unseen manipulations.

1.1 Image Manipulation Detection

The emergence of diverse media tampering tools, such as Photoshop and modern AI-based image editing and generation methods [78, 101, 117, 114, 21], has made it increasingly easy to manipulate visual content. These tools can produce highly realistic edited images that are extremely difficult for humans to detect with the naked eye. While some manipulated images are created for entertainment or artistic purposes, the same accessibility raises serious concerns about the spread of misinformation and the potential security risks that follow. In addition, such undetectable manipulations undermine AI transparency and complicate the responsible use of AI technologies. Therefore, the development and deployment of robust tampering detection techniques—namely, Image Manipulation Detection methods—are essential to mitigate these risks effectively.

Fig. 1.1 illustrates the common high-level pipeline for image manipulation detection and localization. Given an RGB image, the input is passed through an image manipulation detection network, which produces a predicted mask highlighting the potential manipulated regions. Based on this mask, prior methods typically apply a non-trainable pooling strategy to obtain the final image-level prediction indicating whether the image is authentic or tampered. These models are usually trained using a combination of pixel-level losses (supervised by ground-truth manipulation masks) and image-level classification losses.

The task itself consists of two objectives. For a given RGB image, detection determines whether the image has been manipulated (image-level prediction), while localization identifies and delineates the specific manipulated regions (pixel-level prediction). The image-level decision is typically derived from the predicted manipulation mask using pooling operations such as global average pooling. Most prior approaches focus on a small set of classical manipulation types, as shown in Fig. 1.2, including:

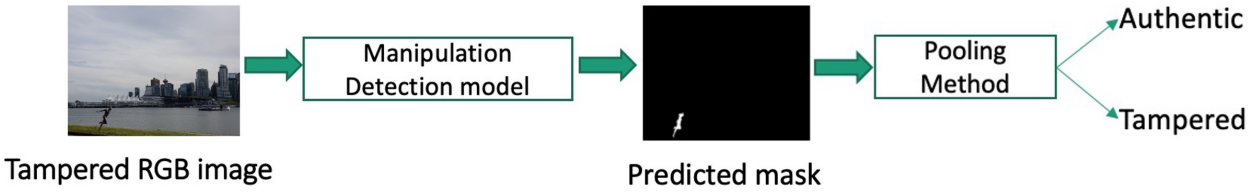

Figure 1.1: The common high-level pipeline of image manipulation detection and localization.

- **Splicing**: inserting content from one image into another to form a composite.
- **Copy–move**: duplicating a region within an image and pasting it elsewhere in the same image.
- **Removal**: deleting a region of an image and filling the missing area using synthesized or inpainted content.

Current state-of-the-art (SoTA) image manipulation detection (IMD) methods can broadly be categorized into two groups: active methods and passive methods. Active IMD relies on pre-embedded signals or watermarks that are intentionally added at the time of image creation. In principle, these embedded markers make manipulation easier to detect. However, active approaches have several major limitations: they require a controlled imaging pipeline in which the camera or generator must insert the marker, and the embedded signal is often fragile, easily degraded by common post-processing operations such as compression or resizing. For these reasons, active methods are rarely used in real-world settings.

In contrast, passive IMD methods do not rely on any pre-inserted information and instead analyze the intrinsic content of the image itself. Based on the level of supervision used during training, passive methods can be further divided into three categories: unsupervised, weakly supervised, and fully supervised approaches. Unsupervised methods [60, 59, 88, 29, 22, 17, 4, 53, 18] typically rely on handcrafted cues such as noise inconsistencies, color filter array (CFA) patterns, or local mosaic inconsistencies. Weakly supervised methods [109] use only image-level labels to identify manipulated images and do not require pixel-level masks. Fully supervised methods [12, 102, 99, 56, 34] instead depend on large-scale datasets containing both image-level and

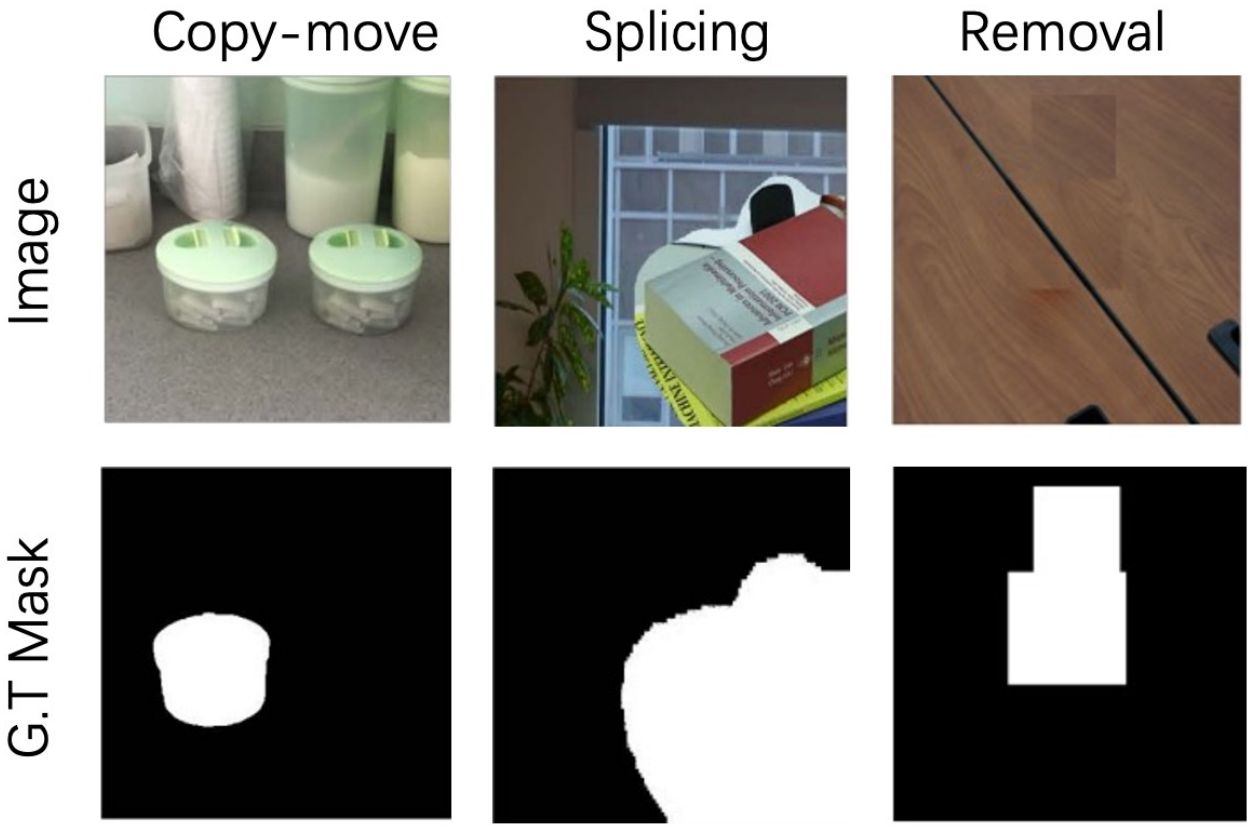

Figure 1.2: Examples of three common image manipulation types: copy–move, splicing, and removal. Image samples are from COVERAGE [95], Colombia [42] and NIST16 [32].

pixel-level annotations, allowing models to learn manipulation traces by exploiting anomalous low-level features. For example, CAT-Net [51] employs a two-branch architecture to jointly detect both editing manipulations and double JPEG compression artifacts.

In general, SoTA works that mention only "detection" in their titles usually address both image-level detection and pixel-level localization, whereas papers that explicitly focus on "localization" tend to restrict their analysis to pixel-level mask prediction alone.

1.2 Motivation of Thesis

Although previous methods have achieved satisfactory results on common manipulation types such as splicing, copy–move, and removal, they often struggle to adapt to manipulations that are more complex or differ substantially from those seen during training. In real-world scenar-

ios, manipulation types are far more diverse, unconstrained, and increasingly influenced by modern AI-based editing tools. Under these conditions, many state-of-the-art methods exhibit significant performance degradation.

Current approaches tend to focus primarily on the three classical manipulation categories introduced in Sec. 1.1 and often rely on strict assumptions about the tampered image. For example, many methods require the manipulated region to be relatively large, or—in the case of JPEG-compressed images—assume that the first and second compression quality factors differ. These assumptions rarely hold in real-world scenarios. When faced with small manipulations, identical recompression settings, or manipulation types outside the conventional categories, existing methods frequently fail to generalize. Consequently, current IMD techniques are not yet equipped to handle the full complexity of real-world manipulation scenarios.

A straightforward way to extend image manipulation detection and localization methods to real-world scenarios is to train models on a wider variety of manipulation types. In principle, exposure to more diverse manipulations could improve a model's generalization capability. However, due to the rapid evolution of editing tools, constructing large-scale datasets that comprehensively represent real-world manipulations is impractical. Such datasets require both accurate image-level labels and precise pixel-level masks—annotations that are extremely time-consuming and costly to obtain. Moreover, inaccurate labels can seriously degrade model performance, making large-scale annotation even more challenging.

Unsupervised manipulation localization methods attempt to avoid these issues by relying solely on handcrafted features such as noise inconsistencies, color filter array (CFA) patterns, or camera fingerprints. However, these approaches typically assume that the input image is already tampered and only aim to highlight possible anomalous regions. As a result, they often produce unreliable or unsatisfactory localization results, especially in modern editing scenarios where manipulation traces may be subtle or semantically driven.

Given the diversity and complexity of real-world manipulations, along with the limited generalizability of existing methods, improving the robustness and adaptability of image manipulation

detection has become both urgent and essential. Accordingly, this thesis focuses on enhancing the generalizability of image manipulation detection methods to better meet the demands of real-world applications.

1.3 Thesis Outline

This section outlines the organization of the thesis and briefly summarizes the content of each chapter.

Chapter 2 reviews the related work, including a detailed description of state-of-the-art methods, the datasets used for image manipulation detection, diffusion models, implicit neural representations and contrastive learning.

Chapter 3 presents the details of our work "A New Benchmark and Model for Challenging Image Manipulation Detection" [121]. In this study, we observe that state-of-the-art methods struggle with more challenging manipulation scenarios, such as extremely small tampered regions and double-compression cases where both compression stages use identical quality factors. We address the limitations of prior fully supervised approaches, which typically rely on restrictive assumptions—namely, that the manipulated region must be sufficiently large and that compressed images must involve different compression quality factors across stages.

We introduce a high-quality benchmark dataset, CIMD, to evaluate state-of-the-art methods under realistic conditions without restrictive assumptions. CIMD contains two complementary subsets: an uncompressed subset designed for image-editing–based detection and a compressed subset targeting double-compression–based methods. The dataset includes fine-grained, high-fidelity annotations at both the image and pixel levels, with images collected under diverse conditions, subjected to complex post-processing, and featuring extremely small manipulated regions.

Unlike previous benchmarks, all images in CIMD contain only tiny manipulated regions, and the double-compressed subset applies identical quality factors in both compression stages.

Our experimental results show that existing state-of-the-art methods perform poorly under these more challenging and realistic conditions, underscoring the need for models that generalize beyond idealized assumptions.

In addition to introducing the benchmark, we propose a novel two-branch model designed to improve detection performance on these challenging manipulation cases without requiring any expansion of the training dataset. This method effectively addresses both tiny manipulated regions and identical–quality-factor double compression, demonstrating stronger generalization capability than prior approaches.

Although the first method presented in Chapter 3 achieves stronger performance on challenging manipulation cases, it remains a fully supervised approach and is therefore limited to detecting common manipulation types rather than novel or unseen ones. To address this limitation, Chapter 4 details our work "Image Manipulation Detection With Implicit Neural Representation and Limited Supervision" [122]. In this study, we introduce a novel image manipulation detection and localization framework that leverages implicit neural representations to support both weakly supervised learning (using only image-level labels) and fully unsupervised learning.

The main objective of this chapter is to explore whether manipulation detection methods can be extended to unseen manipulation types using minimal or no supervision. By relying solely on image-level labels—or, in the unsupervised setting, no labels at all—our method avoids the need to construct large-scale, fully annotated datasets. Such datasets are not only extremely time-consuming to create but are also impractical given the vast diversity of real-world tampering techniques.

In our approach, we use the reconstruction error from the implicit neural representation as the manipulation prior, and we combine this with a novel selective supervision strategy and an adaptive pooling mechanism to enhance generalizability to unseen manipulation types. This framework demonstrates strong potential to detect a broad range of manipulations while substantially reducing dependence on costly dataset construction.

While the framework in Chapter 4 effectively improves the generalizability of image manip-

ulation detection using limited supervision, it still requires model training. Therefore, in Chapter 5, we explore the possibility of a training-free approach for the image manipulation localization task. This chapter presents the details of our work "Training-Free Image Manipulation Localization Using Diffusion Models" [119].

In this study, we leverage the purification capability of diffusion models to develop a fully training-free framework for manipulation localization. Because a pre-trained diffusion model is trained on large-scale clean images, it learns the distribution of authentic images and naturally removes manipulation traces during reconstruction. Unlike conventional approaches, our method requires neither trainable parameters nor dedicated datasets, yet it exhibits strong generalizability to unseen manipulation types in real-world scenarios.

After presenting the details of each proposed method aimed at improving generalizability, Chapter 6 concludes the thesis by summarizing the main contributions, discussing remaining limitations, and outlining promising directions for future research.

CHAPTER 2

Related Works

This chapter reviews the related work relevant to this thesis. The discussion is organized to move from state-of-the-art IMD methods to the core components incorporated in our approaches. Section 2.1 introduces state-of-the-art image manipulation detection techniques. Section 2.2 reviews widely used benchmark datasets that support empirical evaluation and cross-method comparison. Sections 2.3 and 2.4 examine Denoising Diffusion Probabilistic Models (DDPMs) and implicit neural representations, respectively. Finally, Section 2.5 provides an overview of contrastive learning. Portions of this chapter are reproduced from the author's previously published works [121, 122, 119]. Reproduced with permission from the publishers.

2.1 Image Manipulation Detection and Localization

Based on the type of supervision, image manipulation detection methods can be categorized into three groups: unsupervised, weakly supervised, and fully supervised approaches. Unsupervised methods typically rely on hand-crafted features such as noise inconsistency [60, 59, 88], color filter array patterns [29, 22, 17], local mosaic consistency [4], JPEG compression artifacts [53], and camera fingerprints [18]. For weakly supervised learning, [109] introduced a self-consistency learning framework, using FCN [75] as a baseline. Fully supervised IMD methods [12, 102, 99, 56, 34, 7, 45, 33, 91] require large-scale datasets with both image- and pixel-level annotations. Most of these methods learn to identify manipulation traces by modeling anomalous features. For example, CatNet [51] introduced a two-branch network capable of detecting both image editing and double JPEG compression artifacts.

The following provides a brief overview of state-of-the-art image manipulation detection methods:

Unsupervised methods rely on hand-crafted or self-extracted features without any labeled data. Since these techniques generally assume that every input image may contain tampered regions, they are applicable mainly to manipulation localization rather than image-level classification. Early work such as NOI [60] exploits noise inconsistency analysis for blind localization of manipulated areas. CFA [29] leverages color filter array (CFA) pattern analysis to detect regions inconsistent with the camera's demosaicing process. Noiseprint [18] extracts camera-specific fingerprints to reveal local anomalies caused by editing operations. MCA [4] employs an adaptive network to capture mosaic consistency, while IVC [17] applies intermediate-value counting for identifying unnatural transitions introduced by tampering. Similarly, BLK [53] analyzes JPEG compression artifacts to localize inconsistencies within the image block structure.

Weakly supervised methods mitigate the need for dense pixel-level annotations by learning from image-level labels. A representative baseline is FCN [75], which adapts fully convolutional networks to the IMD problem, producing coarse localization maps guided only by global supervision. Building upon this foundation, WSCL [109] introduces a self-consistency learning framework that enforces internal agreement between different image transformations, thereby improving localization accuracy and generalization. These approaches demonstrate that meaningful spatial cues can be learned even in the absence of detailed ground truth masks.

Fully supervised methods constitute the dominant paradigm in recent years, enabled by large-scale datasets providing both image- and pixel-level annotations. RRU-Net [7] first adapts the U-Net architecture with ringed residual connections for more accurate splicing detection. CR-CNN [102] integrates constrained convolution layers to capture noise inconsistencies indicative of manipulation. Mantra-Net [99] learns a manipulation-trace feature extractor jointly with a local anomaly detector, whereas SPAN [45] introduces a pyramid attention mechanism to better aggregate multi-scale contextual information. Extending these designs, PCSS-Net [56] fuses top-down and bottom-up pathways to progressively refine spatio-channel correlations, and CAT-Net [51] learns joint representations of editing and compression artifacts. More recently, MVSS-Net [12] incorporates multi-view and multi-scale supervision by exploiting both noise and edge cues, while TruFor [33] further enhances detection by introducing camera fingerprints as auxiliary input.

Transformer-based architectures such as ObjectFormer [91] and hierarchical frameworks like HiFi-Net [34] represent the latest advancements, offering improved generalization to both traditional and AI-generated forgeries.

2.2 Benchmark Datasets

There are several datasets publicly available that are dedicated to image manipulation detection task. For example, the Columbia Dataset [65] contains uncompressed 363 splicing images of a low average resolution (938×720). CASIA V1.0 and V2.0 [24] were introduced for splicing-only manipulation detection with no ground truth mask. Numerous datasets have been introduced for copy-move tampering detection. For instance, the MICC [1] features images mainly sourced from Columbia photographic image repository. Coverage [95] is another copy-move only dataset includes 100 original-forged pairs with similar-but-genuine objects. The NIST [32] has presented benchmark manipulation datasets with multiple versions. Some large benchmark datasets, such as [61] and [69], apply non-realistic questionable automatically forgeries methods [19] to generate forgery images. In addition, to detect compression artifacts, [51] created five custom datasets that are double compressed using different unreported QFs. [14] proposed a large-scale image manipulation dataset using semantic significance.

2.3 Denoising Diffusion Probabilistic Model

The Denoising Diffusion Probabilistic Model (DDPM) [39] has become popular because of its superior generative capabilities compared to earlier generative models such as GANs [31] and VAEs [48]. DDPM involves two main processes: the forward process adds noise to the image, while the backward process removes the noise to produce a clean image. DDPM has been widely used in media generation and editing [21, 47, 120], image segmentation [96, 2], and image classification [106].

The DDPM [39] consists of two main processes: the forward process, which adds noise, and the backward process, which removes noise. In the forward process, Gaussian noise is gradually

added to the image x_0 to obtain the noised image x_t. The formula for the DDPM forward process is:

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, \epsilon \sim \mathcal{N}(0, I), \tag{2.1}$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$. ϵ is the random noise from normal distribution, and β_t is the predefined variance schedule at a timestep t.

In the backward process, the model removes the noise from x_t to obtain x_{t-1}. The formula for the DDPM backward process is represented as follows:

$$x_{t-1} = \frac{1}{\sqrt{\bar{\alpha}_t}} \left(x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right) + \sigma_t z, \tag{2.2}$$

where $z \sim \mathcal{N}(0, I)$ and $\sigma_t^2 = \beta_t$. $\epsilon_\theta(x_t, t)$ represents the predicted noise of x_t using trained U-Net [80].

The training objective is to minimize the difference between the predicted and ground-truth noise, as shown by the following equation:

$$\mathcal{L}(\theta) = \mathbb{E}_{t, x_{0 \sim data}, \epsilon \sim (0,I)}[\|\epsilon - \epsilon_\theta(x_t, t)\|^2]. \tag{2.3}$$

Classifier and classifier-free guidance: Traditional DDPM often produce random outputs that may not meet specific real-world needs. To address this, conditional DDPM are introduced using either classifier guidance [21] and classifier-free guidance [40]. For classifier guidance, a separate classifier $p(c|x_t)$ is trained to predict a condition c from x_t. Let s_c denote the classifier guiding scale and $\widetilde{\epsilon}(\mathbf{x}_t, c, t)$ denote the conditional output based on condition c on timestep t. The classifier guidance is given by:

$$\widetilde{\epsilon}(\mathbf{x}_t, c, t) = \epsilon_\theta(\mathbf{x}_t, t) - s_c \cdot \sigma_t \nabla_{\mathbf{x}_t} \log p(c|\mathbf{x}_t). \tag{2.4}$$

The main drawback of classifier guidance is the need to train a standalone classifier. To address this, a classifier-free method is introduced in [40]. Let s_f denote the classifier-free guiding

scale. The classifier-free guidance is:

$$\tilde{\epsilon}(\mathbf{x}_t, c, t) = \epsilon_\theta(\mathbf{x}_t, t) + s_f \cdot (\epsilon_\theta(\mathbf{x}_t, c, t) - \epsilon_\theta(\mathbf{x}_t, t)). \tag{2.5}$$

Refer to the original papers [39, 21, 40] for detailed derivation.

2.4 Implicit Neural Representation

Implicit Neural Representation (INR)[13] has become increasingly popular for image modeling. Traditionally, images are treated as discrete signals defined pixel by pixel. In contrast, INR employs a continuous fitting function to represent images, enabling a smooth and flexible representation. The controllable fitting capacity of INR has been widely applied in various tasks, including continuous image and video super-resolution[13, 15], video and image compression [25, 50], continuous shape representation [28, 110], and medical image analysis [64, 111]. More recently, INR has also been adopted for low-light image enhancement [105].

In INR, the input image is first encoded into a feature map $F_N \in \mathbb{R}^{H \times W \times C}$, where H and W denote the height and width, and C is the number of feature channels. The corresponding coordinate set is represented as $X \in \mathbb{R}^{H \times W \times 2}$. The feature map F_N and the coordinate set X are then concatenated and passed through a Multi-Layer Perceptron (MLP) decoder. Formally, the INR process can be expressed as:

$$I_R[x, y] = MLP(F_N[x, y], X[x, y]). \tag{2.6}$$

Here, I_R denotes the reconstructed RGB pixel values of image I, and $[x, y]$ represents the pixel coordinates. The main objective of Implicit Neural Representation (INR) is to recover the RGB values of I, with the loss function formulated as:

$$\mathcal{L}_{INR} = \|I - I_R\|_1 . \tag{2.7}$$

2.5 Contrastive Learning

The core idea of contrastive learning [35, 36] is to push apart negative sample pairs while pulling together positive pairs. One of the most widely used architectures for contrastive learning is the Siamese network [49], which processes two inputs simultaneously and is trained based on their similarity. Contrastive learning has been extensively applied in unsupervised and self-supervised settings [83, 10, 115, 58]. The most commonly used formulation of contrastive loss is the InfoNCE objective [70], defined as:

$$\mathcal{L}_q = -\log \frac{\exp(q \cdot k_+/\tau)}{\sum_{i=0}^{K} \exp(q \cdot k_i/\tau)} \tag{2.8}$$

In the InfoNCE formulation, τ denotes the temperature hyper-parameter. The objective is to bring the positive key k_+ closer to the query while pushing it farther from negative samples. Depending on the specific method and application, alternative formulations of contrastive loss have also been proposed [100, 38, 93].

CHAPTER 3

A New Benchmark and Model for Challenging Image Manipulation Detection

In this chapter, we propose a new benchmark dataset and model for image manipulation detection (IMD) under challenging conditions. Existing IMD methods primarily rely on detecting anomalous features arising from image editing or double-compression artifacts. However, these approaches often impose restrictive assumptions on manipulation conditions—for example, requiring tampered regions to be relatively large or assuming that double compression uses different quality factors. As a result, they struggle to detect manipulations outside these constraints, such as small tampered regions with limited visual information or double-compression cases with identical quality factors. To systematically evaluate state-of-the-art (SoTA) IMD methods under these challenging conditions and to encourage more robust solutions for real-world applications, we introduce the *Challenging Image Manipulation Detection (CIMD)* benchmark dataset. CIMD consists of two subsets tailored to assess editing-based and compression-based methods, respectively, and contains manually captured images that were tampered with and annotated at high quality. In addition, we present a two-branch network model based on HR-Net [89], specifically designed to more effectively detect both editing and compression artifacts in these difficult scenarios. Extensive experiments on the CIMD benchmark demonstrate that our model significantly outperforms existing SoTA IMD methods.

The organization of this chapter is as follows. Section 3.1 motivates the problem and highlights our contributions. Section 3.2 details the proposed CIMD benchmark, including data collection, design choices, and evaluation protocols. Section 3.3 presents the proposed two-branch method and the compression-artifact learning module. Subsequent sections report experimental results and analyses, followed by a discussion of limitations. This chapter is reproduced from [121].

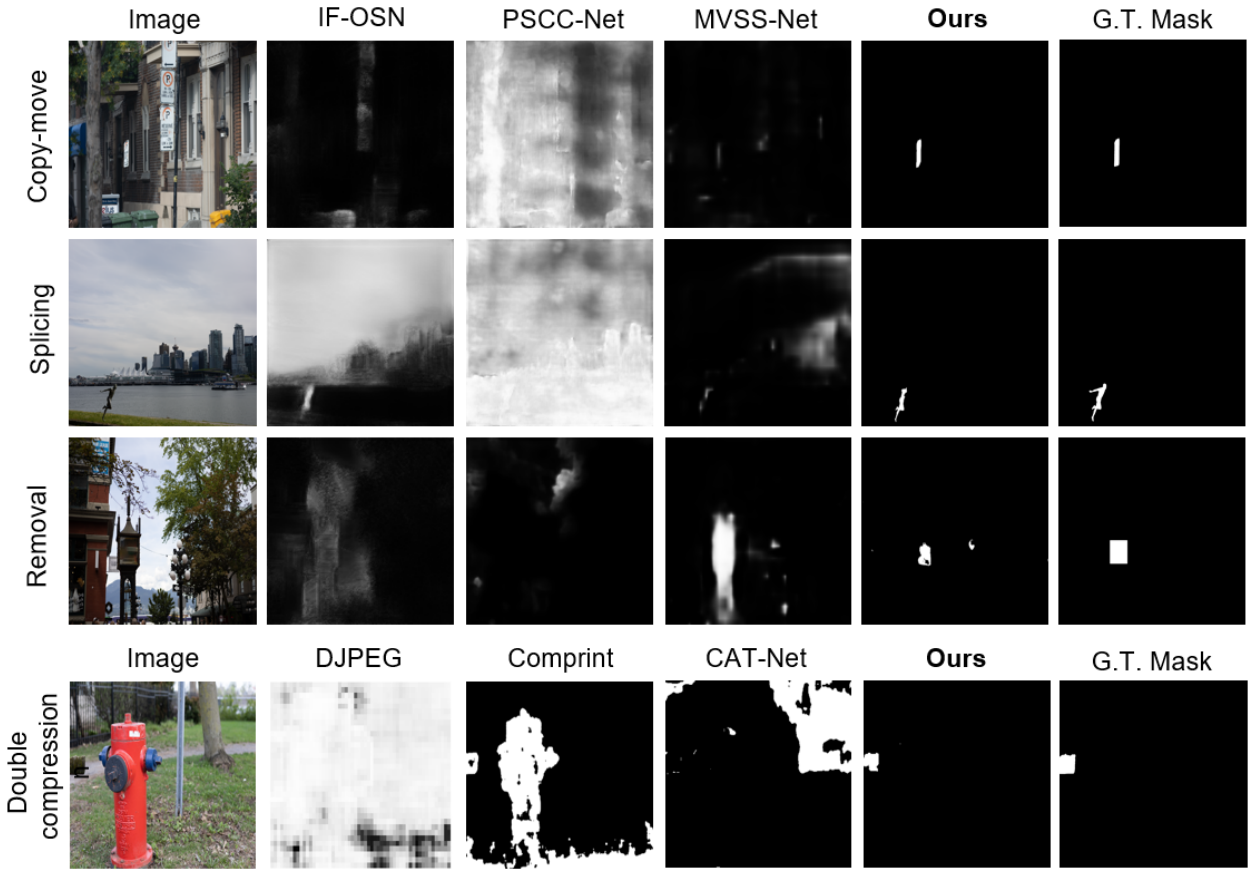

Figure 3.1: Comparison of image manipulation detection performance with recent mainstream methods under challenging conditions. The first three rows show manipulation of region copy-move, splicing and removal, respectively. The last row shows double-compressed splicing with the same Quality Factor (QF). Our method achieves the new state-of-the-art in detecting challenging manipulation cases.

3.1 Motivation and Problem Setting

With the advancement image editing and AI content generation, image editing, tampering and content synthesis are becoming common. However, the abuse of these technologies can bring in serious security and social impacts, including misinformation, disinformation, and deepfakes [44, 85]. **Image Manipulation Detection (IMD)** methods that can accurately detect image manipulation regions are important in media forensics.

There are three general types of image manipulation operations: (1) *region splicing*, where the content from one image is copied and pasted onto another image, (2) *region copy-move*, where an image regions is moved to another location within the same image, and (3) *region removal*,

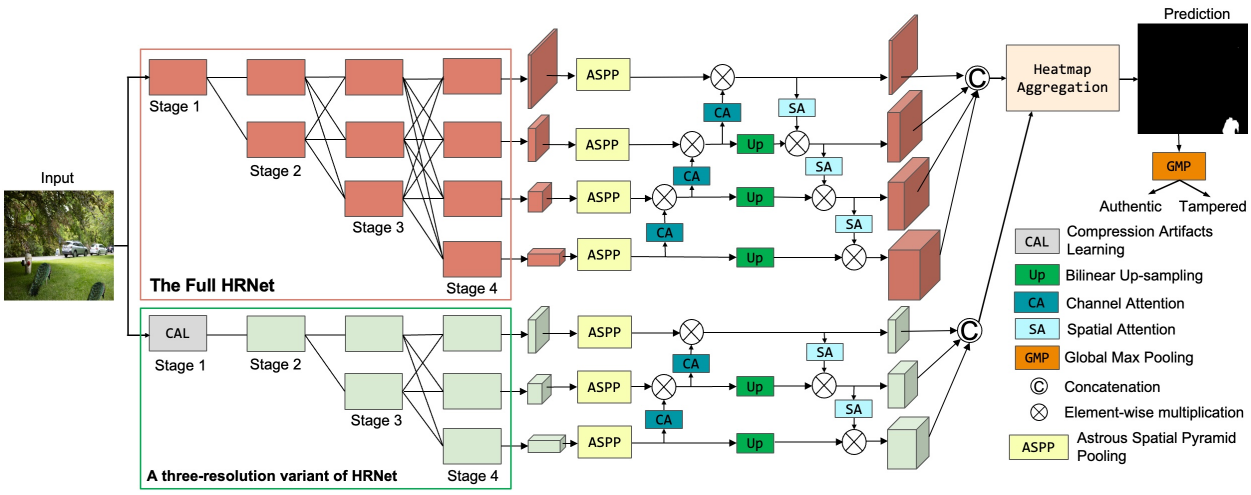

Figure 3.2: Overview of the proposed two-branch architecture. RGB stream can detect anomalous features, while frequency stream is able to learn compression artifacts by feeding the image to the compression artifacts learning model, as depicted in Fig. 3.5. The ASPP in Fig. 3.4(a) is appended to each of the outputs, and channel attention and spatial attention in Fig. 3.4(b)(c) interactively perform between each scale output to improve the detection performance under small manipulation.

where parts of the image are erased and new contents are synthesized. To accurately detect these manipulations, some methods rely on detecting anomalous image region or texture features, while others identify double compression artifacts. While the State-of-the-Art (SoTA) IMD methods perform well on mainstream public IMD datasets, they still face two challenges as shown in Fig. 3.1. First, existing IMD methods have general difficulties in detecting relatively small tampered regions, due to the data-driven design under limited visual information. Secondly, approaches detecting double compression inconsistencies with two different quantization matrices fall apart when the compression Quality Factor (QF) remains the same. This is because the use of identical Q-matrix can significantly suppress double compression artifacts. As shown in Fig. 3.3, methods in this category detect tampered regions by identifying missing histogram values arisen from the two compression processes. When the same QF is used, the histogram undergoes very small changes, making it hard to detect double compression. In summary, as the image tampering techniques improve increasingly fast, forensic problems are typically ill-defined, and IMD methods in general fall behind in research for challenging cases.

To address the issues and challenging conditions, we present a new two-branch IMD network

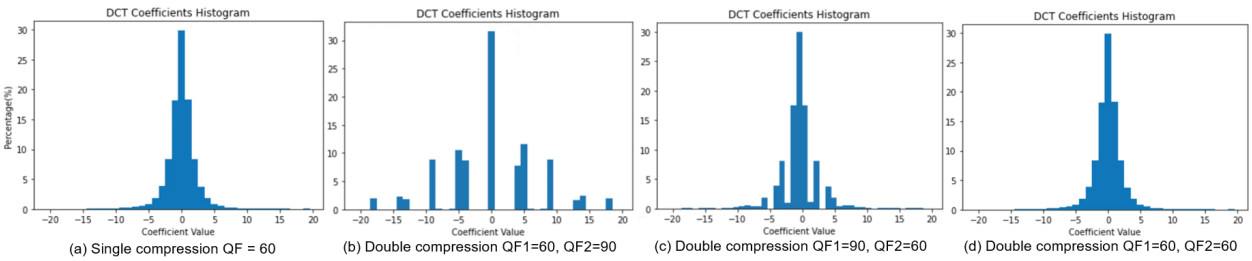

Figure 3.3: DCT coefficient histograms from the (0,1) position generated from a raw image under different compression processes. The range of X-axis is [-20, 20].

incorporating both the RGB and frequency streams, such that both anomaly features and compression artifacts can be detected in a single framework. Our network adopts HR-Net [89] as a feature extractor, with parallel processing at four different scales as in Fig. 3.2. To more precisely pinpoint tiny tampering regions, we carefully designed the model by applying Atrous Spatial Pyramid Pooling (ASPP) [11, 104] and attention mechanism [87, 43]. For the frequency stream, we feed the backbone with quantized DCT coefficients, Q-matrix, and novel residual DCT coefficients from multiple recompressions to detect double compression artifacts. This design works regardless of different or identical QF's. To enhance the performance of the proposed two-branch model, we introduce an adaptive weighted heatmap aggregation design at the end, using soft selection to fuse the heatmaps generated by both branches. Our approach is distinct from the one used in [16], which relies on a simple averaging operation.

Datasets play a critical role in training and evaluating the performance of models. There is no publicly accessible datasets for challenging IMD cases. Existing datasets [24, 95, 65, 32, 1] exhibit a significant imbalance in the distribution of tampered images or contains only one image format, leading to an unreliable measurement of the overall detection capability of models. Additionally, some datasets [61, 69] apply image tampering algorithms *e.g.*, [19] to manipulate images in standard datasets such as MSCOCO [54], which raises concerns, as some IMD methods can rely on MSCOCO pre-trained backbones. In order to evaluate the effectiveness of IMD methods in challenging conditions, we propose a novel **Challenging Image Manipulation Detection (CIMD)** dataset with new features. CIMD consists of two subsets for evaluations of image-editing-based and compression-based methods, respectively.

The primary objective of the first subset is to evaluate the overall performance of image-editing-based methods in detecting small manipulation regions across all three types of manipulations. To ensure fair evaluation, we use raw images without any compression and ensure each type of manipulation contains the same number of samples. The main objective of the second subset is to assess the effectiveness of compression-based methods in detecting compression inconsistency using double-compressed images with identical QF. We created splicing manipulation images in which each double-compressed image was created using the same compression QF from 50-100. CIMD was taken and tampered with manually, ensuring high-quality image samples and annotations. We thus provide a reliable and accurate benchmark for evaluating the performance of image manipulation detection models. The availability of paired authentic and tampered images enables the comprehensive evaluation of a model's ability to identify manipulated images. This chapter makes the following contributions:

- We present a two-branch architecture incorporating RGB and frequency features for challenging image manipulation detection. To our knowledge, our model is the first approach to focus on detecting small tampered regions.
- We introduced the pioneering compression artifacts learning model capable of detecting double-compression artifacts, regardless of whether the quantization factors (QFs) are different or identical.
- We introduce a new high-quality CIMD benchmark for evaluating the performance of SoTA IMD methods in challenging manipulations.
- Extensive experiments on CIMD demonstrate that the proposed approach outperforms the SoTA significantly in challenging image manipulation detection.

3.2 The Challenging Image Manipulation Detection Dataset (CIMD)

The proposed dataset aims to evaluate the performance of models in detecting small region forgeries, which has proven to be a challenging task for current state-of-the-art methods. However,

	CASIA	Columbia (Color)	MICC-F220	CoMoFoD	COVERAGE	NIST16	**Ours**
Year	2013	2006	2011	2011	2016	2016	**2024**
Avg. Resolution	404×321	938×720	550×780	512×512	400×486	2450×3315	2048×1365
Dataset size	800	363	220	260	200	564	**800**
Tampering Type	C,S	S	C	C	C	C,S,R	**C,S,R**
JPEG	✓	-	✓	✓	-	✓	✓
TIFF	Some TIFF	✓	-	-	✓	-	✓
Binary mask	-	-	-	-	✓	✓	✓
Complex Tamper	-	-	-	-	✓	N/A	✓
Var. in Locations	✓	-	✓	✓	-	✓	✓
Small region	-	-	✓	-	-	-	✓

Table 3.1: A comparison of mainstream Image Manipulation Detection (IMD) datasets. The Avg. and Var. are short for average and variation, respectively. Image tempering types of splicing, copy-move, and removal are denoted as S, C, and R, respectively. The entry of N/A means the criteria is not provided in the original paper. CASIA, Columbia, MICC, CoMoFoD, COVERAGE, NIST16, and IMD datasets refers [24], [42], [1], [86], [95], [32], and [69] respectively

among the existing mainstream evaluation datasets, only the MICC dataset [1] focuses on small region forgery with only one type of tampering (copy-move). Therefore, to address this gap in the evaluation, we introduce our new **CIMD dataset**, that is specifically designed to evaluate the detection of small region forgeries. Unlike most previous datasets that only offer images for a specific task, CIMD is a comprehensive dataset that covers multiple types of image forgery tasks, including copy-move, removal, and splicing.

Recent datasets such as NIST16 [32] provide images for the three tampering types (copy-move, removal, and splicing). However, all images in this dataset are compressed using JPEG, which can be unfair in evaluating the performance of some methods. This is because some compression-based methods can detect compression inconsistency instead of image editing traces. Therefore, CIMD contributes to the field by offering both JPEG and TIFF forgery images, enabling fair evaluation of image-editing methods and compression-based methods while eliminating the bias caused by previous datasets with only JPEG images. To achieve a diverse evaluation dataset, we conducted a year-long effort to collect original images, resulting in the first dataset that includes images from all four seasons and different times of day and night.

Table 3.1 provides a comprehensive comparison of mainstream public IMD datasets used in many research papers. In addition to common criteria such as publication year and resolution, our table incorporates criteria such as original image, complex tampering, variation in locations, and all seasons. The original image criterion evaluates whether the datasets employ original, first-hand images as their ground truth photographs. Utilizing first-hand images typically ensures superior image quality and credibility. The complex tampering consists of color and contrast adjustments within the manipulated regions, as these adjustments often yield more convincing forgeries. The variation in location and all seasons in Table 3.1 means that the images were taken at different geographical locations and times throughout the year, ensuring image diversity. Although most datasets introduce image diversity by capturing images at various scenes within a short time frame, they often neglect to consider the influence of seasonal changes. Our dataset addresses this limitation by incorporating longitudinal diversity through the inclusion of images from all four seasons, thereby providing a more accurate representation of daily life.

We aim to build a comprehensive validation dataset (CIMD) dedicated to small region forgery (less than 1.5% on average) in both compressed and uncompressed scenarios. Our dataset are superior in image quality, image diversity, and forgery strategy. Two separate subsets have been introduced to evaluate image editing-based and compression-based methods, respectively.

Collection. We captured original images using Canon RP camera, encompassing both uncompressed TIFF and compressed JPG forgery-original image pairs. These captures were taken across highly diverse multi-season settings, characterized by intricate and sophisticated lighting conditions. Our intention was to offer an impartial and all-encompassing assessment of models within a real-life context. **Two Disentangled Sub-Datasets.** We offer two subsets: the CIMD-Raw subset consists of pairs of original uncompressed TIFF images for the evaluation of image editing-based methods. The CIMD-Compressed subset encompasses splicing forgery and their corresponding original JPEG images with uniform quantization factors (QFs) ranging from 50 to 100. This subset evaluates the capability of compression-based models in detecting forgery under the same QF conditions.

Processing and Tampering. We used Photoshop 2023 (PS) to process and create tampering pho-

tos due to its popularity in other datasets mentioned in the related work section and its popularity in general public.

3.2.1 The CIMD-Raw (CIMD-R) Subset

The CIMD-R benchmark provides a comprehensive evaluation of the image-editing-based models' performance in detecting small tampered copy-move, object-removal, and splicing forgeries on uncompressed images. The use of uncompressed images eliminates undesired compression artifacts on forgery region that can be otherwise sensed by neural networks, enabling a more true performance evaluation on out-of-detection. CIMD-R comprises 600 TIFF images, with a resolution of 2048 $\times$ 1365. Ground-truth masks are also provided. In addition, CIMD-R adopts a future-oriented approach by providing 16-bit image pairs that offer up to 2^{48} (trillions of) colors. For copy-move manipulation, a part of an image is copied and pasted within the same image, followed by five post-processing methods: scaling, rotation, level/curve increasing, illumination changing, and color redistribution. In the case of removal manipulation, forged images are synthesized by removing the selected region from the image (via Content-Aware Fill in PS). Content-Aware Fill is widely used in several datasets [73, 23] and represents the PS's best guess to inpaint the object according to the surrounding region. For splicing forgery, regions from one image are copied and pasted into another source. Then, the same post-processing methods mentioned in copy-move are applied to make the forged region harmonious with its surroundings.

3.2.2 The CIMD-Compressed (CIMD-C) Subset

The CIMD-C benchmark is designed to evaluate the capability of compressed-based models in detecting double JEPG compression artifacts, where the primary and secondary compression has the same QFs. The dataset comprises 200 JPEG images with a resolution of 2048 $\times$ 1365, wherein the QF is uniformly distributed as $50 \leq QF < 100$. Forgery images are generated akin to CIMD-R's splicing samples, with the distinction that the forged image is saved using the JPEG compression algorithm, employing the same QF as the original image. The original images were

produced from RAW files ensuring that the original images are compressed for the first time, enhancing the dataset's credibility. In the forgery images, the background is double-compressed, while the tampered regions are single-compressed. Furthermore, the dataset also comprises binary masks and QF values utilized for compression, thereby augmenting its utility for further investigations into the effects of different QFs.

3.2.3 Ethics Statement

To ensure ethical compliance, all photos presented in our dataset are original and obtained either in public places or with the owners' explicit permission in private places, in accordance with local jurisdiction laws. Moreover, the authors ensure that the photos contain neither identifiable individuals nor personal information. As advised by institutional review boards (IRB), IRB approval is not required for the dataset. Datasets will be publicly available upon acceptance.

3.3 Two-Branch RGB–Frequency IMD Network

The two-branch architecture we propose enables the detection of both anomalous features and compression artifacts inspired by [51]. Furthermore, our model is effective for detecting small manipulation regions and identifying double compression traces that apply the same quantization matrix (Q-matrix). To achieve our research objectives, we adopted HR-Net [89] as the backbone of our model, based on its ability to offer three-fold benefits. Firstly, the absence of pooling layers in HR-Net ensures that the features maintain high resolutions throughout the entire process. Secondly, the model processes features from different scales in parallel with effective information exchange, which is essential for capturing information of varying scales. Finally, the input size of HR-Net is ideally suited for DCT features. Since after processing by dilated convolution with a rate of 8, the size of the DCT feature is reduced to 1/8 of the input size, which is equivalent to the second stage resolution of HR-Net.

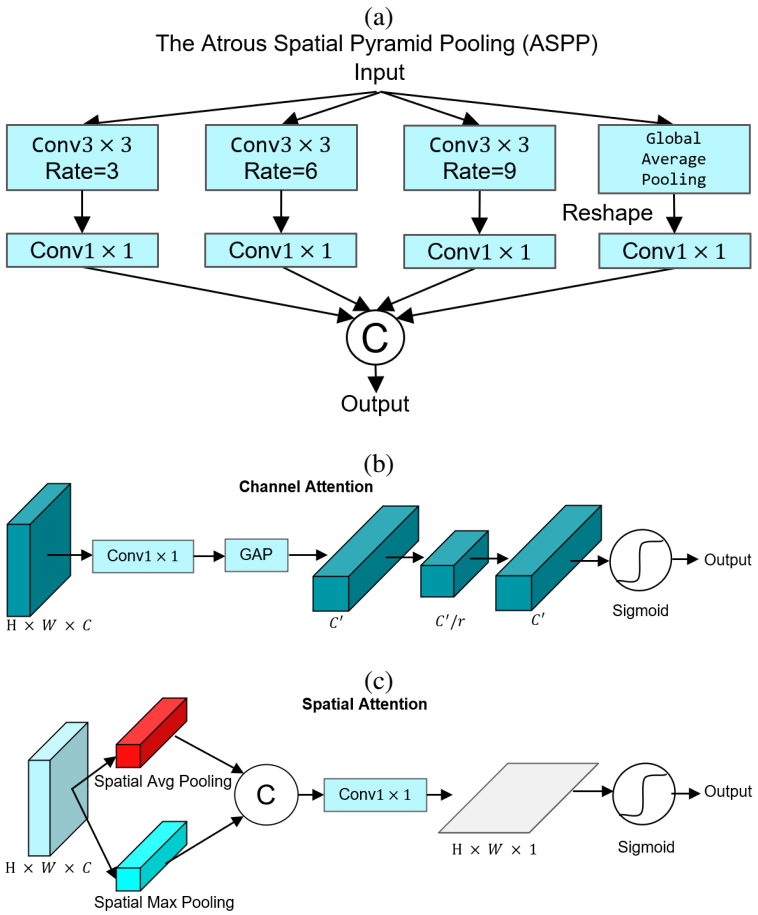

Figure 3.4: Detailed structure of the Atrous Spatial Pyramid Pooling (ASPP), channel attention and spatial attention.

3.3.1 Network Architecture

The network architecture comprises two branches, one for detecting anomalous features and the other for identifying compression artifacts, as in Fig. 3.2. For the RGB stream, the input image is fed to a full HR-Net, which learns the image editing traces from the visual content. In the frequency stream, the image is first input to the proposed compression artifact learning model shown in Fig. 3.5 to extract various DCT features. Subsequently, the DCT features are fed to a variant of the HR-Net, which operates at three different resolutions (1/8, 1/16, and 1/32).

To precisely pinpoint small tampering regions, we carefully designed our model using both Atrous Spatial Pyramid Pooling (ASPP) [11] shown in Fig. 3.4(a) and Attention Mechanism shown in Fig. 3.4(b)(c). The ASPP captures long-range distance information via various receptive fields

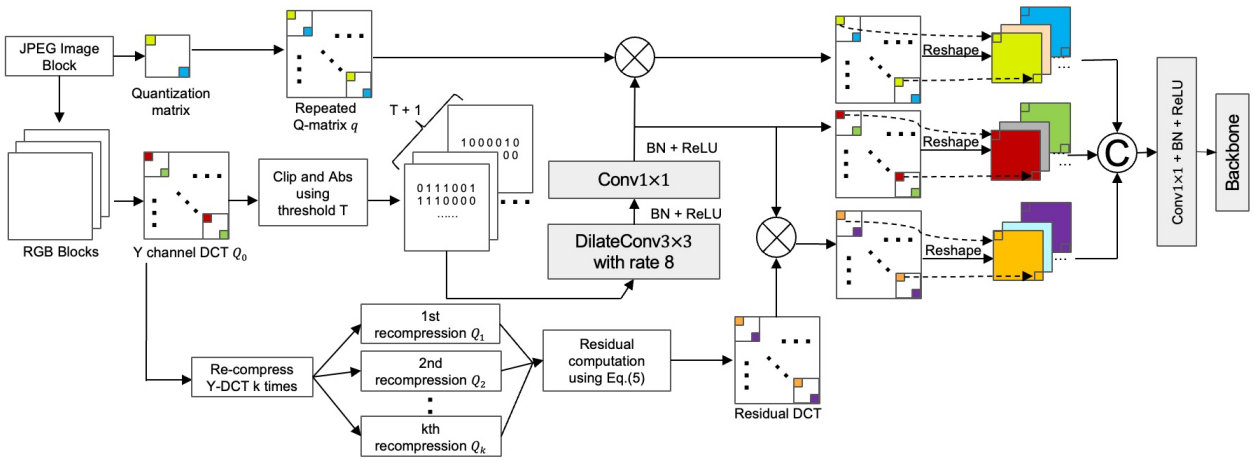

Figure 3.5: The compression artifact learning module. Three types (*de-quantized*, *quantized*, and *residual quantized*) of DCT features are fed into the backbone to learn double compression artifacts in cases whether the QFs are the same or not.

and handles scale variations. It consists of three dilated convolutional layers with different rates and a Global Average Pooling (GAP). The resulting features are concatenated and passed to a 1×1 convolution.

The starting point for designing an attention mechanism between each resolution output of HR-Net lies in the understanding that the four scale features extracted from HR-Net contain a diverse range of semantic and spatial information. Specifically, the high-resolution features contain more spatial content, whereas the low-resolution features carry more semantic responses. However, most prior methods simply do upsampling and concatenate these features for detection without adequately considering their inter-dependencies. The attention mechanism aims to fully leverage the information provided by each resolution and improve detection performance. Specifically, the approach utilizes channel attention from a bottom-up path and spatial attention from a top-down path, where two attention modules collaborate to enhance the features interactively. Through this approach, we seek to fully exploit the potential of each scale feature and improve detection performance.

We next describe how attention works interactively in the RGB stream, where the procedure is virtually identical to the frequency stream, with a different number of output resolution branches. Given a RGB input image I with width W and height H, $I \in \mathbb{R}^{H \times W \times 3}$, the HR-Net output features

in four resolutions can be denoted as $F_1 \in \mathbb{R}^{H/4\times W/4\times C_1}$, $F_2 \in \mathbb{R}^{H/8\times W/8\times C_2}$, $F_3 \in \mathbb{R}^{H/16\times W/16\times C_3}$ and $F_4 \in \mathbb{R}^{H/32\times W/32\times C_4}$, and $C_1 = 48$, $C_2 = 96$, $C_3 = 192$, $C_4 = 384$ as default setting. The bottom-up channel attention feature are calculated using:

$$F_n = \mathcal{C}(F_{n+1}) \odot F_n, \quad n = 1, 2, 3, \tag{3.1}$$

where $\mathcal{C}(\cdot)$ denotes the channel attention block in Fig. 3.4(b) and $\odot$ represents element-wise multiplication. As F_4 contains the highest level of semantic information, it remains unchanged at the channel level.

For the detail of channel attention, the feature maps F_{n+1} undergo an essential preliminary transformation through a 1×1 convolutional layer. This transformation is crucial to ensure that the number of channels between F_{n+1} and F_n is consistent, thereby enabling the element-wise multiplication to be performed effectively in the channel dimension. We set the transformed channel number as $C^{'}$. The transformed features are subsequently fed to a Global Average Pooling, denoted as $GAP(\cdot)$, followed by the excitation process $E(\cdot) = C^{'} \rightarrow C^{'}/r \rightarrow C^{'}$, $r = 4$). The channel attention is calculated as $\mathcal{C}(F) = \sigma\left(E(GAP(Conv_{1\times1}(F)))\right)$, where $\sigma(\cdot)$ is the Sigmoid activation function.

Following the application of bottom-up channel attention, the feature maps F_2, F_3, and F_4 are upsampled using the bilinear upsampling method to match the resolution of F_1. The spatial attention mechanism from the top-down pathway is then applied, which is given by:

$$F_m = S(F_{m-1}) \otimes F_m, \quad m = 2, 3, 4, \tag{3.2}$$

where $S(\cdot)$ is the spatial-attention in Fig. 3.4(c). As F_1 contains the richest spatial information, it remains unchanged at the spatial level. The spatial attention is calculated using the Spatial Max Pooling P_{max} and Spatial Average Pooling P_{avg} as $S(F) = \sigma\left(Conv_{1\times1}\left[P_{max}(F); P_{avg}(F)\right]\right)$, where $[;]$ denotes concatenation.

The feature maps of each branch, after undergoing upsampling and interactive attention,

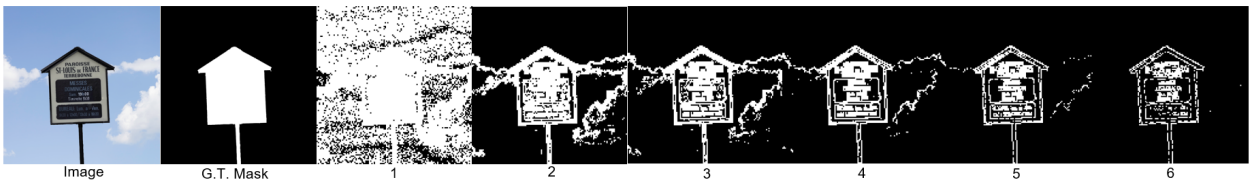

Figure 3.6: Visualization of DCT coefficients for each recompression for a repeatedly compressed image under QF 80. The number below shows recompression counts. Black pixels indicate unaltered DCT coefficients. White pixels indicate the *unstable* region where DCT coefficients change after compression, which gradually focus on the tampered region as the count increases.

have the same resolution. These features are then concatenated together to form final features for adaptive weighted heatmap aggregation in inference stage. Our model generates two final heatmaps, which are aggregated through soft selection. Specifically, we employ bilinear feature upsampling to upscale the heatmap of the frequency stream to match the resolution of the RGB stream heatmap. Following this, we apply the Softmax activation function to the heatmaps, and then use Global Max Pooling (GMP), denoted as $GMP(\cdot)$, to select the main heatmap and its corresponding weight. This selection is based on higher values, which indicate a stronger localization response compared to the other heatmaps. We define the main and secondary heatmap using h_m and h_s. Thus the weighted aggregated heatmap h can be generated using:

$$h = GMP(h_m) \cdot h_m + (1 - GMP(h_m)) \cdot h_s. \tag{3.3}$$

Finally, the same as [12], we apply a non-trainable GMP over the predicted binary mask to perform image-level detection, since image-level detection is highly related to pixel-wise prediction.

3.3.2 JPEG Compression Artifacts Learning Model

Our compression learning model aims to identify compression artifacts in double-compressed images, regardless of whether the primary and secondary compressions have the same QF or not. Several approaches attempt to detect inconsistencies in the DCT histogram, as illustrated in Fig. 3.3(b)(c). It should be noted that when double compression is performed using the same Q-matrix, histogram-based methods are not effective since there are very few compression incon-

sistencies, as shown in Fig. 3.3(d). Fortunately, some traces can still be detected even in such conditions. It was observed in [46] that when a JPEG image is repeatedly compressed using the same QF, the number of different quantized DCT coefficients between two consecutive compressions decreases monotonically. Several methods [76, 103, 67] leverage this evidence to determine whether an image has been single or double-compressed. In contrast to previous approaches, we investigate the feasibility of leveraging this trace to localize tampered regions in an image. Fig. 3.6 shows that when a spliced image is created using the same QF, the manipulated region is singly compressed, however the background regions are doubly compressed. Consequently, when the image is repeatedly compressed, unstable quantized DCT coefficients gradually focus on the tampered area, while the authentic regions remain relatively stable. Based on this observation, we introduce a novel residual DCT map to guide the DCT features to better focus on the unstable regions for IMD.

Our method focuses only on Y-channel DCT map, as it is more sensitive to human eyes. Given a JPEG image, it is easy to obtain the Y-channel quantized DCT coefficients Q_0 and its corresponding Q-matrix from the JPEG file header. The Q-matrix is first repeated to have the same size as Q_0 and we set the repeated Q-matrix as q. Thus, We compute the $(k+1)$th re-compression quantized JPEG coefficients Q_{k+1} using the following equations sequentially:

$$\begin{cases} D_k = Q_k \odot q \\ B_k = IDCT(D_k) \\ I_{k+1} = RT(B_k) \\ Q_{k+1} = [DCT(I_{k+1}) \oslash q] \end{cases}, \tag{3.4}$$

where $\oslash$ denotes element-wise division, D, B, I and Q represent de-quantized DCT coefficients, de-transformed blocks using inverse DCT, image blocks and quantized JPEG coefficients respectively. The subscripts of the variables in the above equations represent the number of recompressions and we experimentally set $k = 7$. $RT(\cdot)$ is rounding and truncation operation. $[\cdot]$ denotes to the rounding operation. Thus, the residual de-quantized DCT coefficients R after k-times recom-

pressions is defined as:

$$R = \frac{1}{k}\sum_{i=1}^{k}(Q_i - Q_{i-1}). \tag{3.5}$$

For original Y-channel DCT coefficients Q_0, we perform a clipping operation using a threshold value T, after which we convert them into a binary volume. Denote this binary value conversion as $f : Q_0^{H\times W} \rightarrow \{0,1\}^{(T+1)\times H\times W}$. It is shown in [107] that f is effective in evaluating the correlation between each coefficient in the DCT histogram. Therefore, the DCT coefficients Q_0 is converted to binary volumes as:

$$f(Q_0^t(i,j)) = \begin{cases} 1, & \text{if } |clip(Q_0(i,j))| = t, t \in [0,T]\,, \\ 0, & \text{otherwise.} \end{cases}$$

The function $clip(\cdot)$ is utilized to extract the histogram feature within $[-T,T]$, which is essential for GPU memory constraints. We set T as 20 from the experiments. Additionally, we apply the absolute operation as DCT histogram exhibits symmetry.

The compression artifact learning method involves two element-wise multiplication operations. The first multiplication is performed between the histogram features and the Q-matrix, which is utilized to simulate the JPEG de-quantization procedure. The second multiplication is used to guide the histogram feature to focus more on unstable coefficients, which is a critical step for detecting double-compressed images using the same QF.

In an 8×8 block of DCT coefficients, each coefficient position represents a specific frequency component. However, the convolution operations in the backbone are designed for RGB images and ignore these frequency relationships. To fully exploit the spatial and frequency information of the DCT coefficients, a reshaping operation is necessary. In detail, each block with a size of $(8\times8\times1)$ is reshaped into a size of $(1\times1\times64)$. Thus, the first and second dimensions represent the spatial information, while the third dimension represents the frequency relationship. Next, the de-quantized, quantized, and residual histogram features are concatenated in the channel dimension. Finally, the concatenated features are input to a 1×1 convolutional layer and the backbone network

for the detection task.

3.4 Experimental Results

We first describe the experimental setup, and then compare the proposed network with the state-of-the-art methods on the newly proposed CIMD dataset.

Datasets. The training datasets used in this study were adopted from [51]. The testing phase entailed the utilization of CIMD-R and CIMD-C to evaluate the efficacy of image-editing-based and compression-based methods, respectively.

Evaluation metrics. Following most previous work, we evaluated the localization results using pixel-level F1 score with both the optimal and fixed 0.5 thresholds. For image-level detection, we employed AUC and image-level accuracy. We set 0.5 as the threshold for image-level accuracy. Only tampered images are used for the manipulation localization evaluation.

Implementation details. Our model was implemented using PyTorch [74] and trained on 8 RTX 2080 GPUs, with batch size 4. We set the initial learning rate as 0.001 with exponential decay. The training process consists of 250 epochs. The proposed model is designed to accept various image formats, including both JPEG and non-JPEG formats. The training objective is designed to minimize the pixel-level binary cross-entropy.

3.4.1 Comparison With State-of-the-Art Methods

To guarantee a fair comparison and evaluate the previous models using newly introduced CIMD, we select the state-of-the-art approaches using these two standards: (1) pre-trained model is publicly available, and (2) the evaluation datasets we used are not in their training sets. Following these criteria, we select RRU-Net [7], MantraNet [99], HiFi_IFDL [34], CR-CNN [102], SPAN [45], PSCC-Net [56], MVSS-Net [12], IF-OSN [98], CAT-Net [51], DJPEG [73] and Comprint [62].

We use CIMD-R to evaluate the performance of the image-editing-based method, while

Method	Pixel-level F1		Image Level	
	Best	Fixed	AUC	Acc
RRU-Net [7]	0.126	0.103	0.500	0.500
CR-CNN [102]	0.126	0.088	0.513	0.502
MantraNet [99]	0.051	0.018	0.500	0.500
SPAN [45]	0.160	0.045	0.510	0.498
HiFi_IFDL [34]	0.145	0.115	0.502	0.502
PSCC-Net [56]	0.208	0.118	0.514	0.505
CAT-Net [51]	0.301	0.194	0.589	0.537
MVSS-Net [12]	0.234	0.153	0.568	0.515
IF-OSN [98]	0.184	0.103	0.516	0.522
Ours	**0.444**	**0.335**	**0.677**	**0.545**

Table 3.2: Evaluation results for image-editing based methods using CIMD-R. Pixel-level F1 scores are calculated using both best and fixed (0.5) thresholds. For image-level performance, AUC and image-level accuracy are reported.

Method	Pixel-level F1		Image-level	
	Best	Fixed	AUC	Acc
DJPEG [73]	0.026	0.022	0.500	0.500
Comprint [62]	0.030	0.010	0.467	0.500
CAT-Net [51]	0.395	0.259	0.534	0.490
Ours	**0.542**	**0.442**	**0.727**	**0.525**

Table 3.3: Evaluation results for compression-based methods on the CIMD-C subset.

CIMD-C is utilized for compression-based approaches.

Evaluation using CIMD-R subset. Table 3.2 reports the results of image-editing-based methods using CIMD-R, in which all image samples are uncompressed. Two Pixel-level F1 scores are calculated using the best F1 threshold for each image and using fixed F1 threshold of 0.5, respectively. Best scores are highlighted in bold. Our method outperforms existing SoTA methods in both image-level and pixel-level evaluation, which demonstrates its superiority for detecting small tampering regions.

Evaluation using CIMD-C subset. Table 3.3 compares the performance of compression-based IMD methods, where all image samples are double compressed using the same QF and the evaluation settings are consistent with those used in Table 3.2. Our method is again the best performer in terms of overall performance, highlighting the effectiveness of our approach for double-compressed images with the same QF.

Method	Columbia		CASIAv1+	
	F1	AUC	F1	AUC
RRU-Net [7]	0.595	0.580	0.410	0.575
CR-CNN [102]	0.436	0.783	0.405	0.719
MantraNet [99]	0.364	0.701	0.155	0.500
SPAN [45]	0.487	0.500	0.184	0.500
PCSS-Net [56]	0.625	0.657	0.520	0.856
CAT-Net [51]	0.876	0.971	0.437	0.647
MVSS-Net [12]	0.655	**0.984**	0.456	0.837
IF-OSN [98]	0.724	0.883	0.509	0.873
Ours	**0.919**	0.910	**0.552**	**0.932**

Table 3.4: Evaluation results compared with SoTA method using two public image manipulation datasets. The evaluation metrics are fixed pixel-level F1 and image-level ROC-AUC.

Evaluation using Public IMD Datasets. We use two publicly available datasets, namely the Columbia [65] and CASIAv1+ [24]), which are widely used in image forgery detection. The performance of our model is compared with existing SoTA approaches on both datasets. Notably, all the images in the Columbia dataset are uncompressed, whereas all samples in the CASIA v1+ are JPEG compressed. This selection allows us to further demonstrate our model's detection ability on both uncompressed and compressed image samples, thereby providing a comprehensive performance evaluation.

Table 3.4 report this evaluation results. Observed that our model demonstrates excellent localization performance on both datasets. In terms of image-level AUC, our model outperforms the SoTA approach on the CASIAv1+ dataset. However, on the Columbia dataset, our model's performance is inferior to that of CAT-Net and MVSS-Net. Our interpretation is that this may be due to the high sensitivity of our model compared to them.

Visualization Results Compared with State-of-the-Art. We provide additional visualizations to compare our approach with existing methods in Figure 3.7 and Figure 3.8. Unlike previous studies that only report tampered detection results, we provide the detection results for each image pair, which includes both the tampered and corresponding authentic image. It is straightforward that when running image manipulation detection on the original unaltered image, any positive re-

Method	CIMD-R Subset		CIMD-C Subset	
	F1	AUC	F1	AUC
RGB Stream	0.330	0.593	0.409	0.525
Frequency Stream	0.130	0.531	0.301	0.512
RGB + Freqnency	**0.335**	**0.677**	**0.442**	**0.727**

Table 3.5: Ablation study of two streams to work collaboratively and/or separately.

sponse reported by the detector is a false positive. We believe that this comparison is more effective in showcasing the overall detection ability of the model for both manipulated and authentic samples. From Figure 3.7 and Figure 3.8, our model achieve the best detection results in both CIMD-R and CIMD-C subsets when compared with the state-of-the-art IMD methods. It is important to note that our method produces not only very good true detections, it is also good in suppressing false positives.

Ablation study. We provide a simple ablation study shown in Table 3.5. Observe that our RGB stream is effective in both compressed and uncompressed data. Notably, the frequency stream fails to produce satisfactory results in CIMD-R due to the absence of compression artifacts. However, when the two branches work collaboratively, the model's performance improves in both localization and detection evaluation.

3.5 Limitation

We provide visualization of some failure cases of our model in Figure 3.9, where our model can not detect the small removal regions. Our interpretation is that this can be due to the insufficient number of removal training samples in our training set, as the CIMD dataset mainly consists of splicing and copy-move manipulation types.

In addition to the bias of training dataset, the new version of Photoshop now provides generative neural networks to perform image inpainting. Image tampering using such technology makes the task of detecting removal region more difficult.

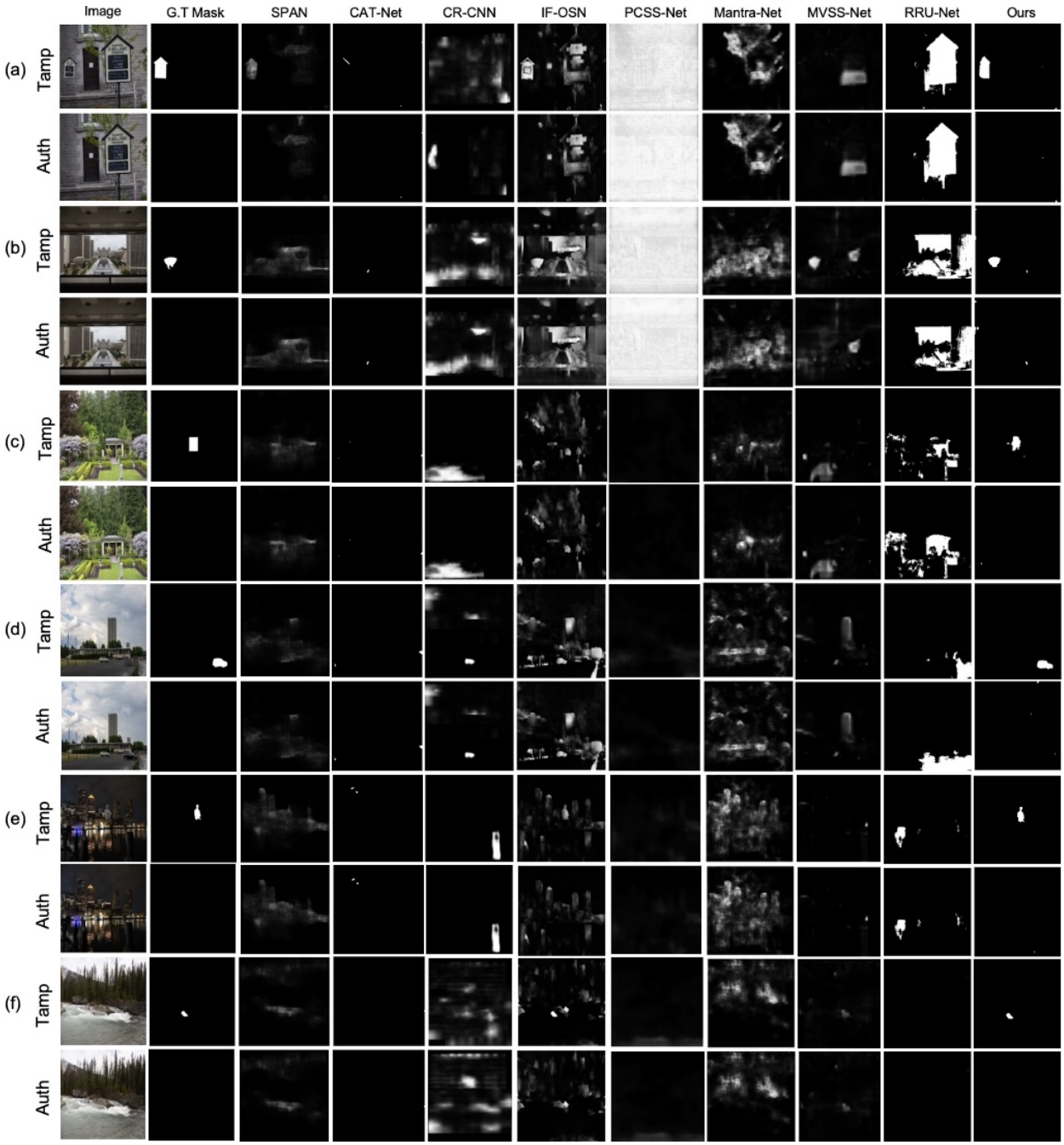

Figure 3.7: Visualization results compared with SoTA image-editing-based methods using CIMD-R subset. We provide the detection results of both tampered and their corresponding authentic images. Each sub-figure in (a-f) contains two rows, where the top row shows the IMD results on a tampered image, and the bottom shows the IMD results running on the unaltered authentic image. We show the IMD results on the unaltered images to highlight the false-positives (FP) of the evaluated methods. Observe that the proposed method has very few FP, showing that it is superior to other methods. The (a-b) input images are tampered with copy-move, the (c-d) input images are tampered with region removal, and (e-f) input images are tampered with region splicing.

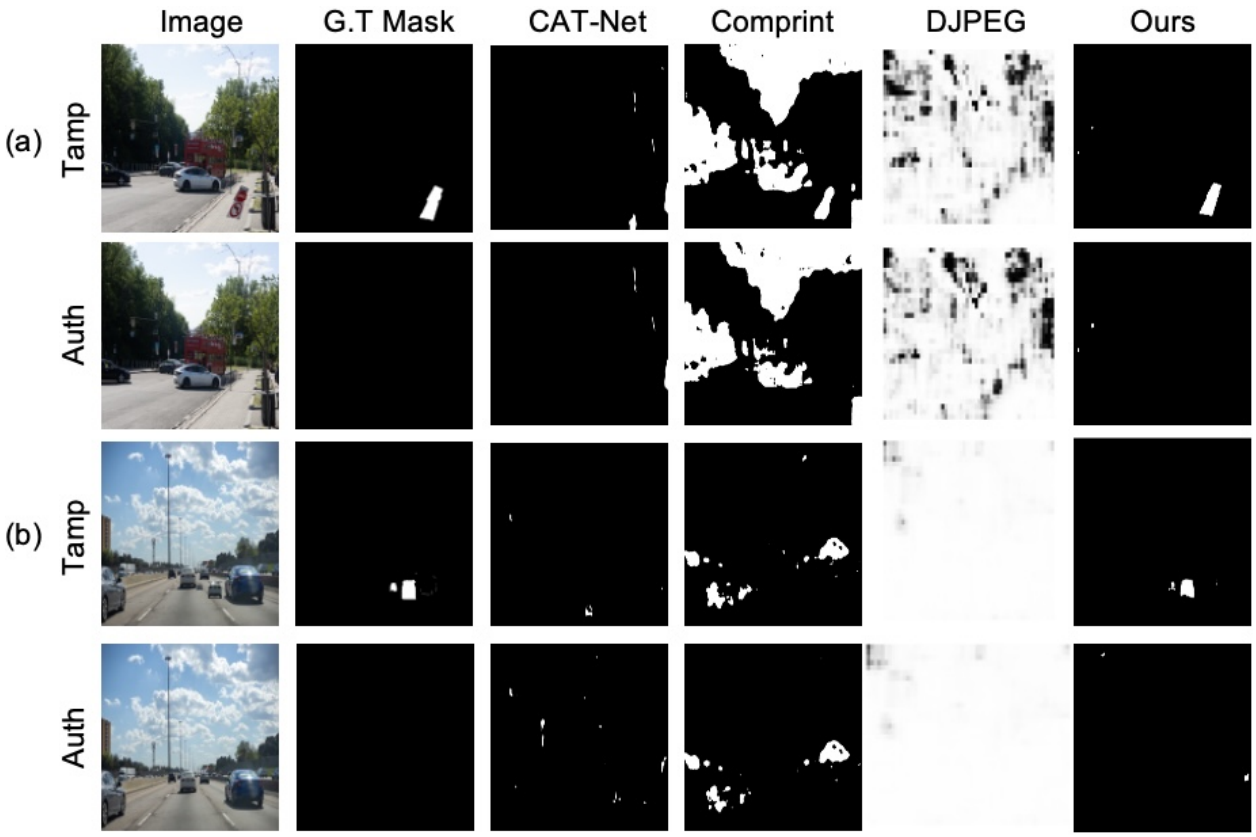

Figure 3.8: Visualization results compared with SoTA compression-based methods using CIMD-C subset. Detection results are provided for both tampered images and their corresponding authentic counterparts. The same as in Figure 3.7, each sub-figure in (a-b) contains two rows, where the top row shows the IMD results on a tampered image, and the bottom shows the IMD results running on the unaltered authentic image.

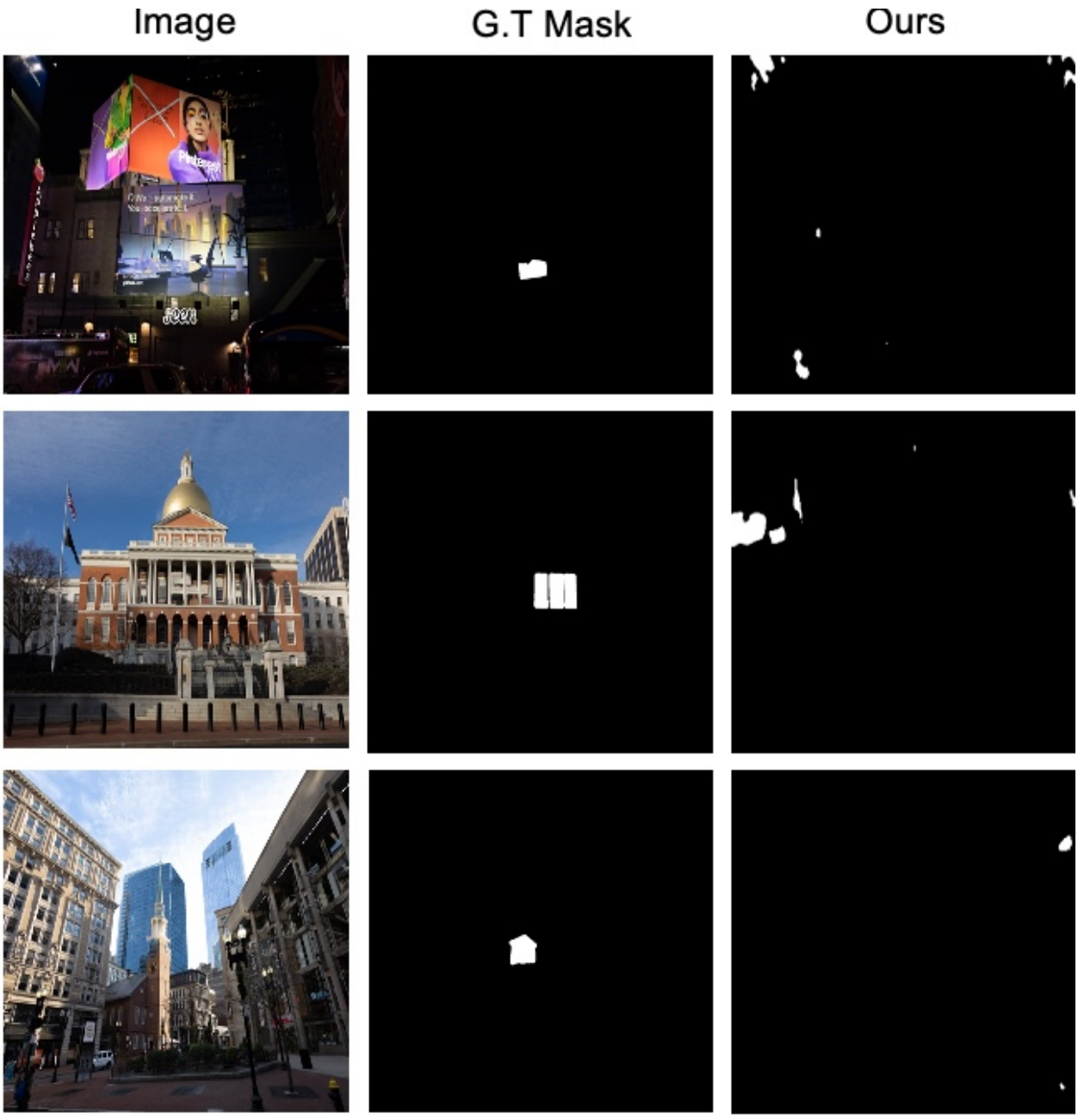

Figure 3.9: Some failure cases of proposed model, where it cannot detect small regions with pixels removed (not copy-move nor spliced).

CHAPTER 4

Image Manipulation Detection With Implicit Neural Representation and Limited Supervision

In the previous chapter, we introduced a new challenging image manipulation detection benchmark (CIMD) along with a method capable of handling such difficult cases. However, that approach is still a fully supervised method, requiring accurate pixel-level annotations. To avoid this dependency and further improve the generalizability of IMD in real-world scenarios, this chapter proposes a novel framework that unifies weakly supervised and unsupervised approaches.

A key limitation of state-of-the-art methods is their reliance on large, high-quality training datasets with both image- and pixel-level annotations. Their performance often degrades when applied to manipulated or noisy samples that deviate from the training distribution. To address these challenges, we introduce a unified framework that integrates unsupervised and weakly supervised strategies.

Specifically, we design a novel preprocessing stage based on a controllable fitting function derived from Implicit Neural Representation (INR). In addition, we propose a selective pixel-level contrastive learning scheme that concentrates on high-confidence regions, thereby reducing uncertainty arising from the absence of pixel-level annotations. In the weakly supervised setting, ground-truth image-level labels guide predictions through an adaptive pooling mechanism. In the unsupervised setting, we adopt a self-distillation strategy that exploits high-confidence pseudo-labels extracted from deep network layers and multiple information sources.

Extensive experiments demonstrate that our framework not only surpasses existing unsupervised and weakly supervised methods but also achieves competitive performance compared to fully supervised approaches on novel manipulation detection tasks.

The chapter is organized as follows. Section 4.1 motivates the problem and presents the

central idea of using INR-based reconstruction errors as manipulation priors, highlighting the key contributions. Section 4.2 details the proposed two-branch architecture, including Neural Representation Reconstruction (NRR), selective contrastive learning, and adaptive global-average pooling (AGAP). Section 4.3 reports experimental results on both standard and novel manipulation benchmarks, followed by ablation studies that validate each component. This chapter reproduces material from [122]. Reproduced with permission from Springer Nature.

4.1 Motivation and Problem Setting

The emergence of diverse media tampering tools, such as Photoshop and AI editing and generation methods [78, 101, 117, 114, 21], has made it increasingly convenient to manipulate media content. However, this accessibility also brings forth the concerning issue of widespread misinformation, which can precipitate serious security implications. Therefore, the development and implementation of robust tampering detection technology, namely, Image Manipulation Detection (IMD) methods, are imperative to mitigate these risks effectively. The fundamental manipulation operations that previous methods typically address are as follows: (1) *Splicing*, which involves taking content from one image and pasting it onto another image, (2) *Copy-move*, in which parts of an image are duplicated and relocated to another location within the same image, (3) *Inpainting*, which entails erasing parts of an image and replacing them with synthesized content.

Despite significant advances in fully supervised IMD methods, they encounter several notable challenges. First, these methods often perform poorly when confronted with unseen manipulation types. Second, extension of them towards unseen manipulation types faces challenges due to their reliance on high-quality training datasets with either image-level and pixel-level annotations. Acquiring such datasets is costly and in many cases, impractical, especially considering the myriad varieties of real-life tampering methods. Third, while some language-guided datasets may lack pixel-level labels, they hold advantages handling real-world scenarios. These datasets can potentially enhance the generalization capability of IMD models.

To address the limitations of fully supervised IMD methods and enhance the generaliza-

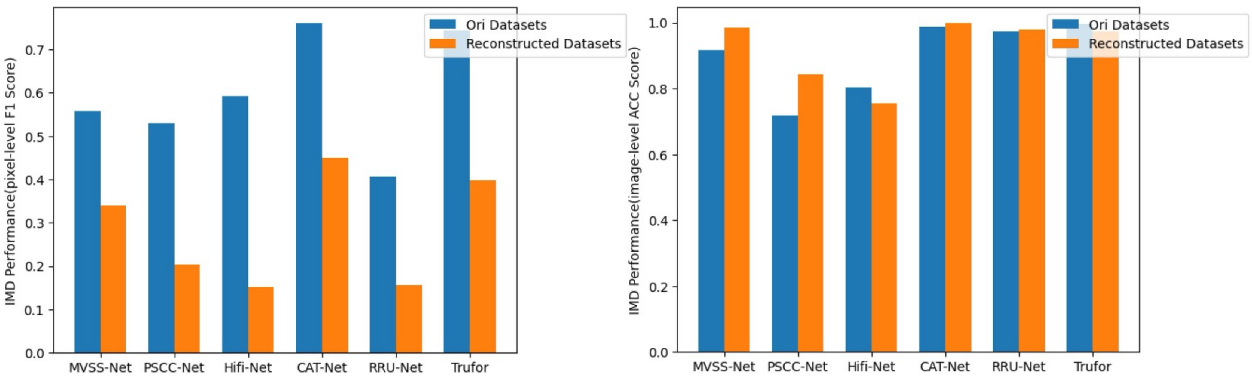

Figure 4.1: We conducted experiments using three widely-used evaluation datasets containing both authentic and tampered samples. Performance are compared with six state-of-the-art fully supervised IMD methods. The pixel-level F1 score is calculated using tampered images, while image-level accuracy is computed using authentic images. The blue and orange bars represent the original datasets and reconstructed datasets via Implicit Neural Representation, respectively. It is evident that there is a significant performance decrease in all methods when applied to reconstructed images in pixel-level detection compared with the original dataset. On the other hand, performance using authentic images shows less change. The scores are averaged across CASIAv1 [23], Coverage [95], and Columbia [42] datasets.

tion ability toward real-world use, we propose to integrate unsupervised and weakly supervised approaches into a unified IMD framework. Our framework allows training with solely image-level labels or even without any labels, aligning with many unsupervised and weakly supervised tasks [116, 26, 118]. Compared to the fully supervised methods, our approach comes with superior generalization capabilities and can be trained using datasets without annotations. Our method begins with the observation that tampered regions exhibit differences from authentic regions in most cases, such as the variations in color and lighting, which pose challenges for the fitting function that needs to model regions accurately. It is shown in [105] that the controllable fitting function of Implicit Neural Representation (INR) tends to learn an average representation of the training images. Motivated by this insight, we raise the following question as our hypothesis: if we train an INR solely on authentic images, can the fitting function effectively represent the characteristics of tampered regions?

To obtain the answer to this question, we first train an INR using only pristine images from CASIAv2 [24] and use it to reconstruct three mainstream datasets. We then apply fully supervised SoTA methods to evaluate the reconstructed datasets, as shown in Fig. 4.1. Surprisingly, the evalua-

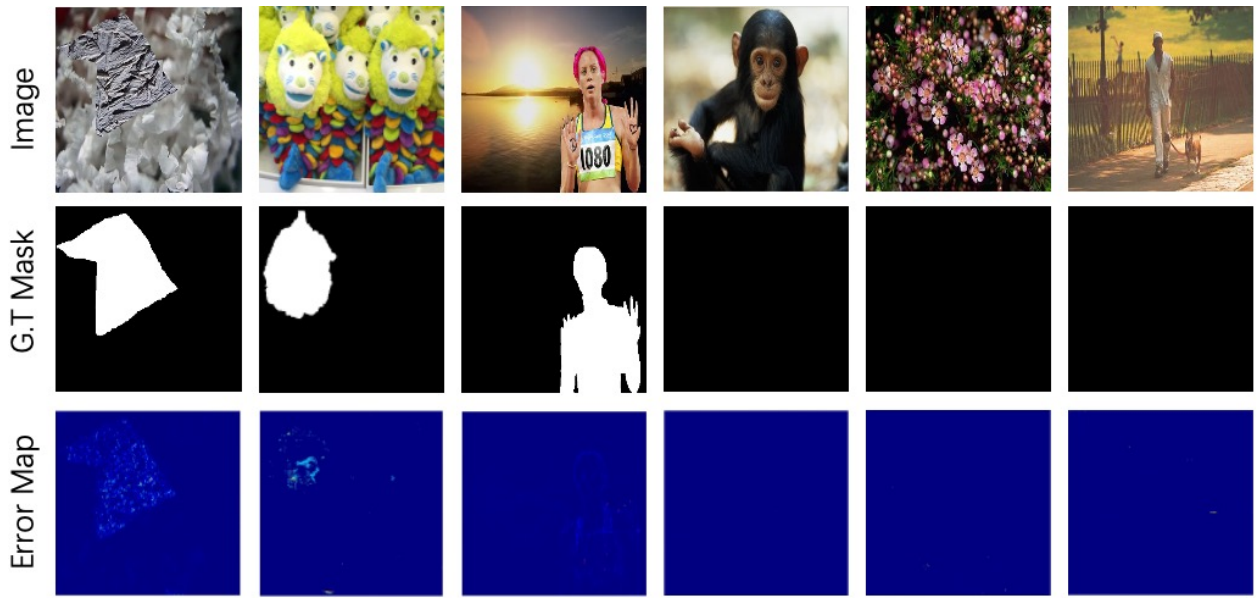

Figure 4.2: Examples of pixel-level Mean Squared Error (MSE) maps computed between original and reconstructed images are presented. The first two rows depict the data samples and their corresponding ground-truth masks, respectively. The first three columns showcase tampered image examples, while the last three columns display authentic images, where the ground-truth masks are all black. Apparently, the reconstruction process fails to properly reconstruct the tampered pixels, resulting in activations in the MSE map. Conversely, there is minimal change observed in the authentic samples.

tion results of these methods exhibit a significant decrease when using INR-reconstructed samples, while there is less performance change in the authentic image samples. This outcome leads us to an initial assumption that the INR may not effectively capture the characteristics of tampered regions. To validate this assumption, we compute the reconstruction error map between the reconstructed and original images in Fig. 4.2. Remarkably, we observe activation in the tampered regions of the tampered samples, while there is no discernible difference in the authentic samples. This observation inspires us to incorporate the INR as a pre-processing method and concatenate the reconstruction error map with the input RGB images before feeding them to the backbone. Notably, this operation alone yields promising results in our weak supervision approach. We name this pre-processing method as **Neural Representation Reconstruction (NRR)**. It is noteworthy that our proposed pre-processing utilizing INR differs from previous methods employing high-pass filters such as SRM [30] or Bayer [102], which can only suppress low-frequency information, thus they are ineffective and inefficient at highlighting potential manipulation regions. Additionally,

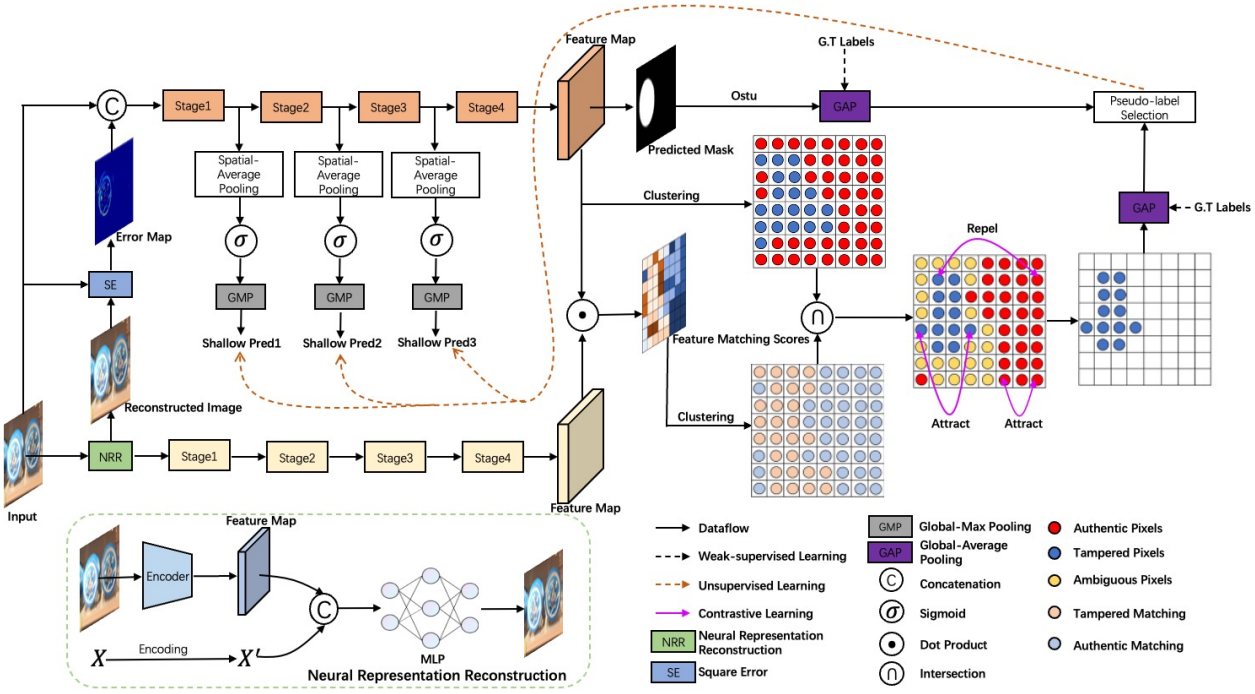

Figure 4.3: An overview of the proposed two-branch framework. The first branch accepts concatenated four-channel inputs as the main branch, while the NRR reconstructed image is fed into the second branch as a complementary branch. Selective contrastive learning is applied only to the pixels that have high confidence of being authentic or tampered. The classification result is conducted by using global-average pooling on both the result of the main branch using Otsu's method and intersected tampered pixels from clustering. In the weakly supervised setting, ground-truth image-level labels are applied for supervision. In the unsupervised setting, high-confidence pseudo-labels from the deepest layers are used to guide the shallow outputs.

they can only work when tampered parts originate from a different source, such as splicing or removal. Our method, however, proves effective even when the tampered parts originate from the same source as the authentic parts, such as copy-move operations.

Following the success of the pre-processing using INR, we further explore our findings and leverage it fully in our framework. Drawing inspiration from Contrastive Learning [36], we utilize NRR as a contrastive sample generator and introduce selective pixel-level contrastive learning, focusing solely on highly confident regions. This approach effectively mitigates uncertainty associated with the absence of pixel-level labels and further enhances weakly supervised performance. We further extend our method to a fully unsupervised approach trained with selected high-confidence pseudo-labels using a self-distillation [113] training strategy. Finally, previous

SoTA methods widely apply Global-Max Pooling (GMP) or Global-Average Pooling (GAP) for image-level detection. However, GMP can hinder training and cause inaccurate predictions, as only the most discriminative response is back-propagated, neglecting the entire tampered content. Conversely, GAP is susceptible to inaccuracies due to weakly activated pixels. To overcome this limitation, we propose an adaptive global-average pooling that focuses on the high-confidence tampered regions. Our method can thus produce more comprehensive and robust image-level predictions.

Experimental evaluations are conducted on seven datasets, including five mainstream datasets featuring general manipulation types and two novel datasets containing unseen tampered samples. The results demonstrate that our methods outperform SoTA weakly and unsupervised methods. Furthermore, our method achieves competitive results compared to fully supervised methods in novel manipulation detection tasks. Finally, our method can be easily extended to the datasets without pixel-level labels, which shows enhanced generalizability.

This chapter makes the following contributions:

- We propose a novel method that achieves plausible weakly and unsupervised IMD results. Our method can be easily adapted to images without labels or only with image-level labels.
- To our knowledge, we are the first to investigate the potential of Implicit Neural Representation (INR) in the IMD task. The pre-process step utilizing INR demonstrates effectiveness in handling tampered cases.
- We introduce selective supervision, which mitigates uncertainty associated with the absence of labels and further improves detection performance.
- Extensive experiments validate the efficacy of our proposed methods, showcasing superior performance on both standard and novel manipulation types compared to SoTA methods.

4.2 INR-Guided Weakly and Unsupervised IMD Framework

4.2.1 Overall Architecture

Fig. 4.3 illustrates the overall architecture of our IMD framework. The basic architecture comprises two branches with shared weights. Given an RGB image $I \in \mathbb{R}^{H\times W\times 3}$ where H and W are its height and width, respectively, we first apply Neural Representation Reconstruction (NRR) to reconstruct it as $I_R \in \mathbb{R}^{H\times W\times 3}$ and generate a reconstruction error map $I_E \in \mathbb{R}^{H\times W\times 1}$ between I_R and I. We then concatenate I and I_E, feeding them into the first branch, which serves as the main branch. Similar to most IMD methods, the main branch generates a mask using a simple upsampling and Sigmoid activation function on the final feature map. We then apply Otsu's method to adaptively select the activated region for image-level prediction, as done in [109]. The reconstructed image I_R is fed into the second branch, acting as a complementary branch for feature matching. After processing with the backbone, we obtain two feature maps F and F_R. We next compute the feature matching scores M between the two feature maps via a dot product, where authentic pixels tend to have higher matching scores and vice versa. For the two-class classification of manipulation detection, unsupervised clustering is applied to F and M. We then intersect the two clustering results and exclusively apply pixel-level contrastive learning to the intersected features that exhibit higher confidence in being either authentic or tampered. The image-level classification result is conducted using the proposed adaptive global average pooling, which focuses on the high-confidence tampered regions for comprehensice image-level prediction. In a weakly supervised manner, the ground-truth image-level label is applied to supervise the prediction. In an unsupervised manner, a selected set of high-confidence pseudo-labels from the deepest layer are utilized to supervise the shallow prediction via self-distillation [113] training strategy. The high-confidence pseudo-labels are chosen by comparing predictions derived from Otsu's method and clustering technique, opting solely for those consistently identified by both sources.

4.2.2 Neural Representation Reconstruction

Inspired by [105] and the observation from our experiment in Fig. 4.1 and 4.2, we apply NRR to reconstruct the input image. The reconstruction error can highlight the manipulation trace, thereby furnishing an indispensable prior to the subsequent IMD model. In INR, the input image is first converted to a feature map $F_N \in \mathbb{R}^{H\times W\times C}$ using an image encoder, where H and W are height and width, C is the number of feature channels. The coordinate set of input can be expressed using $X \in \mathbb{R}^{H\times W\times 2}$. We proceed by concatenating F_N and X, subsequently feeding them into a Multi-Layer Perceptron (MLP) for decoding. The NRR is formulated as:

$$I_R[x, y] = MLP(F_N[x, y], X[x, y]), \tag{4.1}$$

where I_R is reconstructed RGB pixel values from I, $[x, y]$ is each pixel location. The main goal of NRR is to reconstruct the RGB values of I, with loss function formulated as:

$$\mathcal{L}_{NRR} = \|I - I_R\|_1 . \tag{4.2}$$

Note that such reconstruction can not depict the high-frequency pixels properly. We therefore apply the positional encoding from [63] to map X to a higher-dimensional space. Such positional encoding is expressed as:

$$X^{'} = \left(\sin(2^0\pi X), \cos(2^0\pi X), \cdots, \sin(2^{L-1}\pi X), \cos(2^{L-1}\pi X)\right), \tag{4.3}$$

where L is a pre-setting constant to control the fitting ability of NRR. Normally, larger L results in a more accurate fitting. In our task, we aim to avoid outputs from NRR that mirror the inputs; instead, we desire NRR to faithfully preserve information in normal (authentic) content while introducing unfaithfulness in extreme (tampered) pixels. We empirically choose $L = 8$ as the optimal trade-off.

4.2.3 Selective Contrastive Learning

After obtaining I_R from NRR, we calculate the reconstruction error map between I and I_R using $I_E = (I_R - I)^2$. Then, we concatenate I_E and I, enhancing the input to the backbone's first (main) branch. For the input of the second (complementary) branch, we send I_R for feature matching. We employ ResNet50 [37] as a backbone, which consists of four stages that match previous weakly supervised methods. The weights of two branches are shared. After processing by the backbone, we obtain 2 feature outputs F and F_R from different input sources. We then compute the feature matching scores M using the dot product as:

$$M_{x,y} = \sigma\left(\frac{P(F_R^{x,y}) \cdot P(F^{x,y})}{\sqrt{C}}\right), \tag{4.4}$$

where $M_{x,y}$ is the similarity score at the spatial location (x, y). The project head $P(\cdot)$ contains 2 convolutional layers and ReLU activation. The $\sigma(\cdot)$ denotes the sigmoid activation function, and $\sqrt{C}$ provides normalization.

Due to the ability of NRR to properly reproduce only authentic pixels (and not tampered ones), the high matching scores in M tend to correspond to the authentic parts of the image. In contrast, low scores tend to correspond to manipulated regions of the image. Due to the lack of ground-truth masks to supervise the final features, we apply unsupervised clustering for forged/pristine classification similar to [97, 8, 68, 72, 77] and assume that the cluster with fewer elements is the tampered cluster. This assumption aligns with the real-world situation of current manipulation datasets. The reason is that, in most cases, the tampered region is usually much smaller than the authentic ones.

Ideally, we can apply pixel-level contrastive learning through InfoNCE [36] on M and F as [97]. However, we found that this method does not work well in our experiments, as clustering may come with low confidence due to the lack of ground-truth masks. To address this issue, we intersect on the clustering results of M and F, and denote the intersected clustering as C_I. After the intersection, we will have 2 clusters with higher confidence in being either authentic or tampered with, since they come from the same prediction from 2 different sources. We thus apply InfoNCE

only to intersected pixels for contrastive learning, leaving the ambiguous pixels unchanged. This selective contrastive learning loss is formulated as:

$$\mathcal{L}_{SCL} = -\log \frac{\frac{1}{J}\sum_{j\in[1,J]} \exp(q \cdot k_j^+/\tau)}{\sum_{i\in[1,K]} \exp(q \cdot k_i^-/\tau)}, \tag{4.5}$$

where q is an encoded query; J and K are the number of selected positive and negative keys, respectively; τ is a temperature hyper-parameter. We set positive keys k_j^+ as pixels associated with pristine regions, whereas negative keys k_i^- correspond to pixels linked to tampered regions.

4.2.4 Adaptive Global Average Pooling

Many existing methods use Global-Max Pooling (GMP) and Global-Average Pooling (GAP) for image-level prediction to determine if the input is authentic or tampered. However, GMP can hinder training and cause inaccurate predictions as only the most discriminative response is back-propagated, neglecting the entire tampered content. Global-Average Pooling (GAP) is susceptible to inaccuracies due to weakly activated pixels.

To tackle these challenges, we introduce **Adaptive Global Average Pooling (AGAP)**, which focuses on the high-confidence tampered regions for comprehensive image-level prediction. Leveraging the intersection of two clustering results (discussed in Section 4.2.3), we initially apply Global Average Pooling (GAP) exclusively to intersected tampered regions from a clustering perspective. However, relying solely on unsupervised clustering may not guarantee optimal performance and robustness across all input types without ground-truth labels. As discussed in [55], Otsu's method performs well when the image histogram exhibits a bimodal distribution, whereas clustering provides flexibility and the ability to handle more complex histograms. Therefore, we combine Otsu and clustering to enhance image-level prediction and training robustness. Specifically, GAP is applied to the tampered responses from both Otsu and intersected clustering results for loss computation with image-level labels. Further details on Otsu's method and clustering can be found in their respective papers [71, 27].

4.2.5 Weakly-supervised and Unsupervised IMD

In the *weakly-supervised* IMD setting, we utilize ground truth image-level labels to supervise the prediction training using a binary cross-entropy (BCE) loss, which is:

$$\mathcal{L}_{BCE}(g, \hat{g}) = -(1 - g)\log(1 - \hat{g}) - g\log(\hat{g}), \tag{4.6}$$

where g and $\hat{g}$ are the ground-truth and prediction scores, respectively. The final classification loss in a weakly supervised manner is the sum of two BCE losses, comparing two pooling results with g.

In the *unsupervised* IMD setting, where no labels are used, we employ a self-distillation training strategy [113], using pseudo-labels from the deepest layers as a teacher to supervise the shallow outputs.

To streamline prediction results from shallow layers and mitigate computational overhead, the classification head following each middle stage of the backbone uses spatial-average pooling in the channel dimension, reshaping it into a one-channel feature map. This is followed by a sigmoid function and global-max pooling. In traditional self-distillation methods, combining ground truth loss and self-distillation enhances overall performance, but this approach is not applicable in an unsupervised context. Our experiments revealed that relying solely on self-distillation did not yield satisfactory results, as the outputs from the deepest layers may lack accuracy, hindering the training process and overall performance.

Drawing inspiration from the selective-supervised method [52], proven effective in handling noisy label datasets, we leverage its concept of selecting training examples based on the alignment between feature representation and given labels. However, in our unsupervised setting, the absence of labels poses a challenge. To overcome this hurdle, we compare predictions obtained from Otsu and clustering methods, choosing only those consistently predicted by both sources as pseudo-labels for self-distillation training.

In pseudo-label selection, predictions exceeding 0.5 are considered as tampered samples.

Similar to weakly supervised setting, we employ BCE loss between selected pseudo-labels and shallow predictions for supervision. During inference, all classification heads in shallow layers are excluded to avoid unnecessary parameters.

Training objective. We first apply trained NRR through $\mathcal{L}_{NRR}$ as a pre-trained model, with all its weights frozen during IMD training. For simplicity, we use the symbol $\mathcal{L}_{cls}$ to denote the loss functions for classification in both unsupervised and weakly supervised approaches, albeit with slight differences as described above.

The total loss for our proposed IMD, denoted as $\mathcal{L}_{total}$, is a weighted sum of both classification losses using BCE loss and the selective pixel-level contrastive learning loss:

$$\mathcal{L}_{total} = \alpha\mathcal{L}_{cls} + \beta\mathcal{L}_{SCL}, \tag{4.7}$$

where α and β are weighting hyperparameters.

4.3 Experimental Results

4.4 Experiments

In this section, we provide details on the experimental setups, present the comparison results with state-of-the-art (SoTA) methods, and conduct some ablation studies.

Dataset: Our model is trained using CASIAv2 [24] exclusively, which comprises 7,491 authentic samples and 5,063 tampered images. For the evaluations of the standard IMD task, we employ widely-used benchmarks, including CASIAv1 [23], Coverage [95], Columbia [42], IMD2020 [69], and NIST16 [32]. CASIAv1 [23] consists of both splicing and copy-move images without ground truth. Coverage [95] contains only copy-move samples with some post-processing. Columbia [42] comprises 363 uncompressed images with an average resolution of 938×720. NIST16 [32] and IMD2020 [69] contain only tampered images, suitable for pixel-level evaluation. These datasets cover traditional manipulation types including splicing, copy-move, and removal.

For evaluations involving novel or more complex manipulation types, we utilize IEdit [81] and MagicBrush [112], which are two language-driven datasets containing various novel manipulation types, such as action change and light change.

Table 4.1 provides details of the training and testing datasets. Instances marked as N/A indicate either a lack of information in the original paper or the presence of tampered types distinct from the conventional three types.

Split	Dataset	Authentic	Tampered	Copy-Move	Splicing	Removal
Training	CASIAv2 [24]	7,491	5,063	3,235	1,828	0
Testing	CASIAv1 [23]	800	920	459	461	0
	Columbia [42]	183	180	0	180	0
	Coverage [95]	100	100	100	0	0
	NIST16 [32]	0	563	68	288	208
	IMD2020 [69]	0	2,010	————N/A————		
	IEdit [81]	401	445	————N/A————		
	MagicBrush [112]	535	535	————N/A————		

Table 4.1: Comparison of datasets used for training and testing.

Evaluation Metrics: We utilize IOU and F1 scores, including P-F1 for pixel-level F1, I-F1 for image-level F1, and C-F1 for combined F1. The C-F1 score accounts for both pixel-level and image-level performance through the harmonic mean, providing an overall performance comparison. All F1 scores and IOU scores are computed using 0.5 as the fixed threshold. Due to the lack of pixel-level masks in IEdit [81], we include image-level ACC for additional evaluation.

Implementation Details: We employ ResNet50 [37] as the backbone and the model is implemented using PyTorch [74], with parameters initialized randomly. We apply AdamW [57] as the optimizer. The Multi-Layer Perceptron (MLP) in NRR follows a three-hidden-layer architecture. NRR is trained for 120 epochs with an initial learning rate of 2×10^{-4} and weight decay is applied. The IMD model in weakly supervised mode is trained for 50 epochs with an initial learning rate of 0.0005 and weight decay. For the unsupervised model, we train for 20 epochs with an initial learning rate of 0.0001, applying weight decay. Image augmentation is limited to random flipping and cropping.

Method	CASIAv1		Columbia		Coverage		NIST16		IMD2020	
	IOU	P-F1	IOU	P-F1	IOU	P-F1	IOU	P-F1	IOU	P-F1
NOI [60]	0.075	0.132	0.152	0.236	0.122	0.210	0.048	0.074	**0.091**	0.126
CFAl [29]	0.081	0.134	0.175	0.275	0.103	0.185	0.076	0.105	0.068	0.103
MCA [4]	0.049	0.089	0.085	0.148	0.078	0.136	0.049	0.074	0.044	0.079
NoisePrint [18]	0.074	0.130	0.085	0.320	0.098	0.176	0.062	0.106	0.054	0.104
IVC [17]	0.056	0.101	0.085	0.164	0.070	0.127	0.038	0.068	0.048	0.086
Ours	**0.097**	**0.166**	**0.216**	**0.344**	**0.131**	**0.217**	**0.080**	**0.129**	0.079	**0.136**

Table 4.2: Evaluation results of unsupervised methods for Standard Manipulation task.

Method	CASIAv1				Columbia				Coverage				NIST16		IMD2020	
	IOU	P-F1	I-F1	C-F1	IOU	P-F1	I-F1	C-F1	IOU	P-F1	I-F1	C-F1	IOU	P-F1	IOU	P-F1
FCN [75]	0.078	0.122	0.561	0.200	0.062	0.098	0.524	0.165	0.072	0.122	0.424	0.190	0.032	0.052	0.052	0.086
WSCL [109]	0.100	0.163	0.679	0.263	0.220	0.321	**0.720**	0.444	0.102	0.171	0.571	0.263	0.047	0.078	0.093	0.152
Ours	**0.124**	**0.199**	**0.703**	**0.310**	**0.248**	**0.365**	0.695	**0.479**	**0.140**	**0.221**	**0.667**	**0.332**	**0.079**	**0.131**	**0.124**	**0.204**

Table 4.3: Evaluation results of weakly supervised approaches for Standard Manipulation task.

4.4.1 Comparison with SoTA Methods

For a fair comparison with state-of-the-art (SoTA) methods, we selected approaches for which the source code is publicly available. Among the unsupervised methods applied for comparison are NOI [60], CFAl [29], MCA [4], NoisePrint [18], and IVC [17], while the weakly supervised methods include FCN [75] and WSCL [109].

Additionally, we conducted experiments using two novel manipulation datasets and compared our approach with fully supervised methods, including RRU-Net [7], Mantra-Net [99], SPAN [45], PSCC-Net [56], Trufor [33], CAT-Net [51], Hifi-Net [34], CR-CNN [102], ObjectFormer [91], and MVSS-Net [12].

Comparison with SoTA unsupervised methods: Due to the assumption of unsupervised methods that all images contain manipulated parts, they will classify all test images as tampered. Thus, they are not suitable for image-level evaluation. We conduct pixel-level experiments to compare their abilities to localize the manipulated region, as shown in Table 4.2. We can observe that our proposed method in the unsupervised setting achieves the best detection performance compared to other unsupervised methods across five widely used standard manipulation benchmarks.

Type	Method	Training Data Size	IEdit		MagicBrush			
			ACC	I-F1	AUC	P-F1	I-F1	C-F1
Full Supervision	RRU-Net [7]	4.2K	0.482	0.651	0.536	0.153	0.667	0.249
	Mantra-Net [99]	64K	0.499	**0.665**	0.497	0.105	0.667	0.181
	SPAN [45]	96K	0.528	0.210	0.494	0.002	0.585	0.004
	PSCC-Net [56]	100K	0.524	0.206	0.493	0.210	<u>0.710</u>	0.324
	Trufor [33]	858K	0.505	**0.665**	**0.617**	**0.304**	0.670	**0.418**
	CAT-Net [51]	858K	0.488	0.567	0.509	0.033	**0.766**	0.063
	Hifi-Net [34]	1,710K	0.531	0.460	0.457	0.151	0.677	0.247
	CR-CNN [102]	12.5K	0.531	0.530	0.500	0.042	0.593	0.078
	ObjectFormer [91]	12.5K	0.497	0.427	0.514	0.047	0.430	0.085
	MVSS-Net [12]	12.5K	0.526	0.487	0.516	0.072	0.675	0.130
Weak	FCN [75]	12.5K	0.481	0.220	0.506	0.035	0.360	0.064
	WSCL [109]	12.5K	0.511	0.475	0.538	0.122	0.572	0.201
	Ours	12.5K	**0.535**	<u>0.664</u>	<u>0.611</u>	<u>0.264</u>	0.690	<u>0.382</u>

Table 4.4: Evaluation results on Novel Manipulation task for both fully supervised and weakly supervised methods. For the methods that are trained on a dataset with a size of 12.5K, they all utilize CASIAv2 [24] as the training set. For the methods not utilizing CASIAv2, except for RRU-Net, they utilize their synthesis datasets. The best and second-best performances are highlighted using bold and underline, respectively.

Comparison with SoTA weakly supervised methods: Table 4.3 shows experimental results comparing weakly supervised SoTA approaches. Except for the F1 (I-F1) score at the image level in the Columbia [42] dataset, our method performs better than state-of-the-art methods in all other metrics. Regarding the relatively lower I-F1 score in Columbia compared to WSCL [109], we believe the reason is that Columbia does not have post-processing, so our method may not be very sensitive to manipulation. However, despite this issue, our method achieves the best localization performance in the Columbia dataset.

Comparison using novel manipulation dataset: In order to show the generalization ability of our method. We conduct evaluations using fully supervised and weakly supervised methods on two novel manipulation detection datasets in Table 4.4. We can see that the fully supervised methods cannot adapt to the novel manipulation type, resulting in low detection performance even if they utilize a very large synthesis training dataset with both image-level and pixel-level labels. In contrast, our method achieves competitive performance while using extremely few training data with only image-level labels.

Visualization Results: We present some visualization results compared to SoTA methods in

Method	CASIAv1			NIST16
	P-F1	I-F1	C-F1	P-F1
Baseline [75]	0.070	0.363	0.117	0.084
Baseline+NRR	0.123	0.517	0.199	0.114
Baseline+NRR+SCL	0.156	0.591	0.247	0.119
Baseline+NRR+SCL+AGAP	**0.199**	**0.703**	**0.310**	**0.131**

Table 4.5: Ablation study of proposed components using CASIAv1 and NIST16 on Weakly Supervised setting.

Method	CASIAv1 P-F1	NIST16 P-F1
w/o PLS	0.132	0.082
w/ PLS	**0.166**	**0.129**

Table 4.6: Ablation on pseudo-label selection.

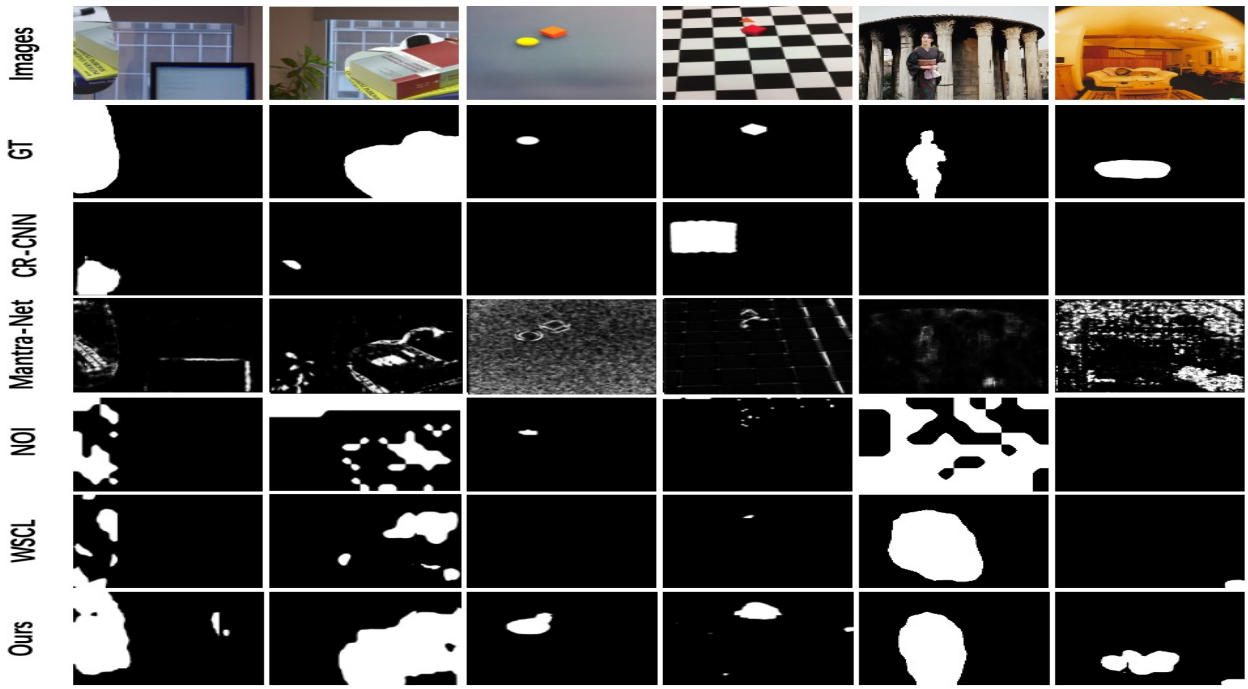

Figure 4.4: Visualization results using different methods. The images are displayed in the following order from top to bottom: tampered images, ground truth masks, prediction results from CR-CNN, Mantra-Net, NOI, WSCL, and our method.

Figure 5.3. Our method can better localize the tampered region, even without the use of pixel-level labels. However, due to the lack of pixel-level labels, our model cannot accurately detect tampered edges. These results of our method are generated from the weakly supervised model, and more visualization results are provided in the additional material.

Method	CASIAv1			NIST16
	P-F1	I-F1	C-F1	P-F1
Global-Max Pooling	0.033	0.570	0.062	0.058
Global-Average Pooling	0.158	0.256	0.195	0.081
Generalized Mean Pooling [79]	0.067	0.626	0.121	0.072
Global Smooth Pooling [92]	0.076	0.627	0.136	0.080
Adaptive Global-Average Pooling (Ours)	**0.199**	**0.703**	**0.310**	**0.131**

Table 4.7: Comparisons using different pooling methods.

4.4.2 Ablation Study

We conducted several ablation studies to evaluate the effectiveness of each proposed component. For these studies, we utilized the CASIAv1 [24] and NIST16 [32] datasets.

Effectiveness of proposed components: We introduced three novel components: pre-processing stage using Neural Representation Reconstruction (NRR), Selective Pixel-wise Contrastive Learning (SCL), and Adaptive Global-Average Pooling (AGAP) for both un-/weakly supervised IMD. The ablation study conducted in weak mode is shown in Table 4.5. It is evident that with the progressive integration of our proposed modules, the model's overall ability to detect tampering consistently improves.

Pseudo-Label Selection (PLS): In our unsupervised method, we introduce PLS, which exclusively leverages high-confidence pseudo-labels from two sources to supervise shallow predictions in the self-distillation training process. The impact of PLS is examined in Table 4.6. In experiments without PLS, we use image-level predictions from the mask in the main branch as pseudo-labels to guide shallow predictions. The proposed PLS proves to be effective in enhancing unsupervised performance.

Adaptive Global-Average Pooling: To demonstrate the superiority of the proposed AGAP, we conduct an ablation study in weakly supervised setting using different pooling methods, including Global-Max Pooling (GMP), Global-Average Pooling (GAP), Generalized Mean Pooling (GeM) [79], and Global Smooth Pooling (GsM) [92]. The results are shown in Table 4.7. Similarly, the proposed AGAP achieves the best performance, highlighting its superiority.

CHAPTER 5

Training-Free Image Manipulation Localization Using Diffusion Models

In the previous chapter, although we proposed weakly supervised and unsupervised methods that reduce reliance on labels, these approaches still require training. In this chapter, we introduce a training-free method for image manipulation localization (IML) that requires neither model training nor annotated datasets. This work is motivated by the limitations of existing IML methods, which typically depend on extensive training with both image-level and pixel-level annotations and often struggle to generalize to unseen manipulation types.

To address this issue, we propose a training-free approach based on diffusion models. Specifically, our method adaptively selects an appropriate number of diffusion timesteps for each input image in the forward process, and then performs both *conditional* and *unconditional* reconstructions in the backward process without relying on external conditions. By comparing these reconstructions, we generate localization maps that highlight manipulated regions through their inconsistencies.

We evaluate our approach against sixteen state-of-the-art (SoTA) methods across six benchmark datasets. Experimental results demonstrate that our method not only surpasses existing unsupervised and weakly supervised techniques, but also achieves competitive performance compared to fully supervised methods on novel, unseen manipulation types.

The remainder of this chapter is organized as follows. Section 5.1 motivates training-free IML, defines the problem setting, and highlights the key contributions. Section 5.2 presents the full pipeline of the proposed method. Section 5.3 reports extensive experiments on six benchmarks against sixteen baselines, including unsupervised, weakly supervised, and fully supervised methods. This chapter is reproduced from [119]. Reproduced with permission from the publisher.

5.1 Motivation and Problem Setting

Image manipulation localization (IML) aims to locate tampered regions within an image. This technology has become increasingly important due to the advancements in media editing and generation methods, such as Photoshop and Generative AI techniques [78, 101, 117, 21], to ensure media authentication. Traditional image manipulation types fall into three categories: *removal*, where media content is removed and synthesized; *splicing*, which involves inserting content from a different source into an image; and *copy-move*, which involves relocating content within the same image.

Even though fully-supervised IML methods have achieved satisfactory localization performance on some common IML datasets, they still have several drawbacks. First, they require extensive training with datasets including image and pixel-level annotations, which are costly. Second, these methods perform poorly when localizing manipulation types different from those in the training datasets, resulting in low generalizability and unsatisfactory performance in real-world scenarios. Given the numerous and ever-growing types of tampering, it is impractical to create datasets that fully encompasses all tampering types for model training.

To address the aforementioned issues as well as improve the generalizability of IML methods for real-world scenarios, this work explores the possibility of a training-free method for IML that does not require any training datasets for learning. Our initial experiment is inspired by **diffusion purification** methods [66, 90], which have demonstrated that *diffusion models (DM)*, having learned the clean data distribution, can effectively remove adversarial attacks. As an extension of image purification, the work of [82] shows that DM can hide manipulation traces, resulting in decreased performance of IML methods. Based on this idea and its successful results, we propose the following hypothesis: *Since DM learns the clean data distribution using authentic images, and forensic traces can be hidden after the diffusion reverse process, can we use this property to locate possible manipulations through the reconstruction inconsistencies?*

To verify this hypothesis, we start with feeding tampered images into an unconditional DM, akin to the diffusion purification, to obtain the reconstructed images. A key issue before this is

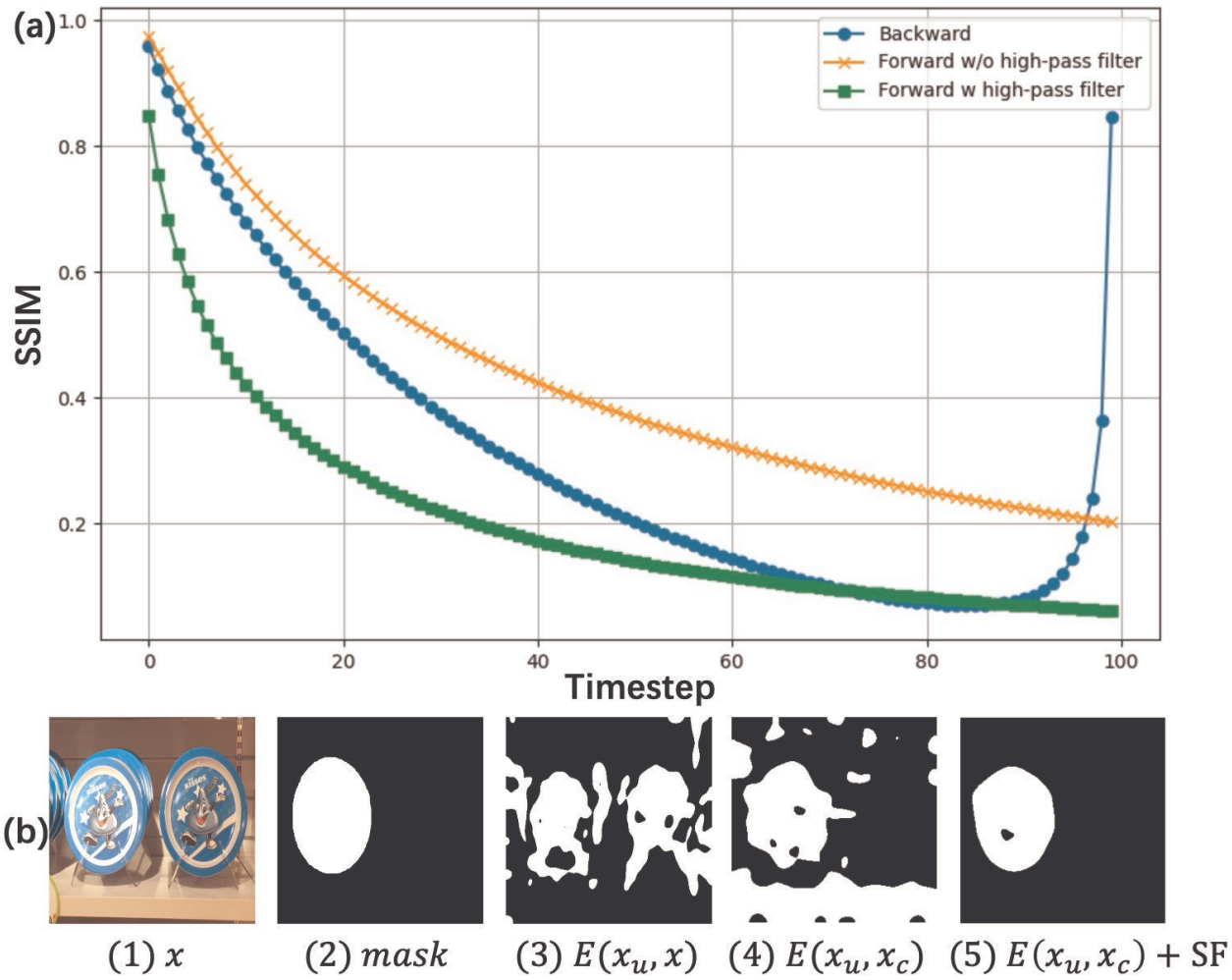

Figure 5.1: (a) SSIM scores at various timesteps are shown for forward and backward diffusion processes. For the forward process, results with a high-pass filter are indicated by a green line, and without a high-pass filter by an orange line. The backward diffusion process is depicted with a blue line. These scores are averaged across CASIAv1 [23], Coverage [95], and Columbia [42] datasets. (b) From left to right: the tampered image, ground-truth mask, and three error masks (unconditional reconstruction *vs.* input, unconditional *vs.* conditional reconstruction, and unconditional *vs.* conditional reconstruction with self-attention guidance).

choosing the appropriate number of *diffusion timesteps* T. If T is too large, the reconstructed image may deviate significantly from the input, introducing unwanted artifacts. Conversely, if T is too small, the method might not effectively remove the tampered traces. Previous purification methods often use a fixed T for all images, which is clearly suboptimal. Inspired by the high-pass (HP) filters [30, 5] commonly used in IML to enhance performance by filtering out image content, we use HP filters to assess whether tampering traces have been effectively removed. The green and

orange lines in Fig. 5.1(a) illustrate the Structural Similarity Index Metric (SSIM) [94] at different timesteps in the diffusion forward process, with and without the application of HP filters. The SSIM scores are calculated between each time-step-noised sample and the original input. Both trends show a consistent decrease, indicating that more noise leads to greater deviation from the original input. The SSIM with HP filters (green curve) drop more rapidly, which helps in selecting T that effectively removes tampered traces while preserving the structure of the input image.

We obtained the reconstruction error map by comparing the original input against the reconstructed image. Unfortunately, the results did not align with our expectations and assumptions, as shown in Fig. 5.1(b3), where the error map covers the entire foreground region. This observation shows that using DM directly cannot differentiate between the tampered and authentic image regions. The underlying issue is that while the DM can reconstruct the tampered image to align with a clean distribution, leading to inconsistencies in tampered regions, it fails to accurately reconstruct the authentic pixel values, resulting in unexpected inconsistencies in the authentic regions as well. To address this issue, we modify the diffusion reverse process to start from the same noised image x_T and perform both **conditional** and **unconditional** reconstructions. The conditional reconstruction is guided by the forged image, using similarity scores SSIM [94] to reconstruct the tampered traces, while the unconditional reconstruction generates a clean image devoid of manipulation traces. We seek for a diffusion reconstruction that minimizes inconsistencies in authentic pixels, while ensuring that the error is concentrated solely on the tampered regions, such that IML can be achieved. We also ensure that both reverse diffusion processes use the same random noise in the sampling step to minimize the impact of noise randomness of the results.

The error mask between two backward processes focuses more on the tampered region rather than the entire foreground, as shown in the example in Fig. 5.1(b4). However, there is one more challenge to overcome: Due to the global guidance of the conditional generation by SSIM, the result still contains many false alarms. To address these false positives, inspired by the *self-attention (SF) guidance diffusion model* [41], which demonstrates that self-attention masks from DM overlap with high-frequency regions. We incorporate the guidance from both SF and SSIM into the conditional branch to direct the reconstruction more precisely towards the tampered regions. The

final error mask, shown in Fig. 5.1(b5), contains much less false positives, achieving the best IML effects. Unlike traditional guided diffusion methods, our conditional backward process does not require any external conditions (such as class labels or text), thereby demonstrating strong generalizability.

We also observed that the SSIM values between unconditional and conditional samplings along backward timestamps, shown by the blue curve in Fig. 5.1(a), initially decrease and then increase. This pattern is similar to what was observed in [9] with external conditions (image-level labels). Based on this observation, and following the approach in [9], we obtain the final error mask by aggregating the error maps starting from the backward timestep when SSIM reaches its minimum. This approach produces the best performance in our experiments. Fig. 5.2 overviews our method.

We evaluated our IML method on six public datasets: five standard datasets with common tampering types and one novel dataset with unseen and more complex manipulation types. The results show that our training-free method outperforms State-of-The-Art (SoTA) unsupervised and weakly-supervised approaches. Additionally, our method competes effectively with fully-supervised methods on unseen, novel manipulation types, demonstrating stronger generalizability.

The contributions can be summarized as follows:

- We present a novel image manipulation localization approach that does not require any training or training data.
- The conditional backward process in our method operates without relying on external conditions, making the approach more generalizable.
- We conducted comprehensive evaluations of sixteen SoTA methods using six IML datasets, encompassing unsupervised, weakly-supervised, and fully-supervised approaches. The results demonstrate superior performance on both standard and novel tampered datasets compared to existing SoTA methods.

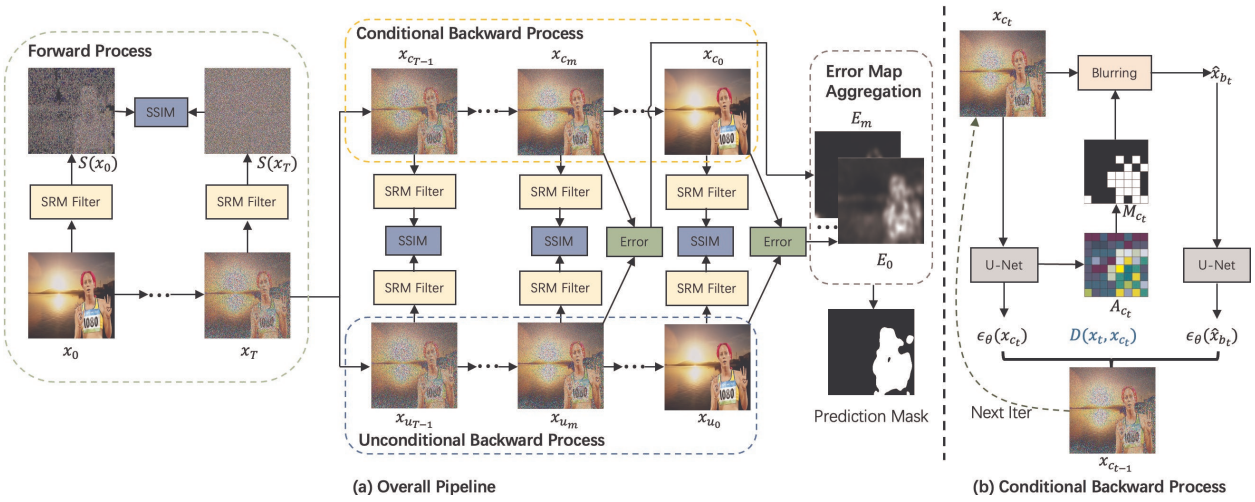

Figure 5.2: (a) Overview of our IML method. In the forward process, $S(x_0)$ is compared with each $S(x_t)$ using SSIM scores. These scores help choose the appropriate T to remove manipulation traces while preserving the input image's structure. Two backward processes then aggregate the error maps starting from the backward timestep m, where SSIM is lowest. (b) The conditional denoising is guided by both self-attention and similarity.

5.2 Training-Free Diffusion-Based Localization Pipeline

Fig. 5.2(a) shows the overall pipeline of our method, which is both training-free and condition-free. Our method includes a single *forward process* that adds noise to the image and two *backward processes* that reconstruct the image with and without manipulation traces. Let t denote a timestep where $t \in [0, T]$, with T being the final timestep. During the forward process, samples are denoted as x_t. In the *conditional* and *unconditional* backward processes, samples are denoted as x_{c_t} and x_{u_t}, respectively. Let A_{c_t} denote the self-attention map, and M_{c_t} denote the corresponding attention mask.

In the forward process, let $S(\cdot)$ denote the steganalysis rich model (SRM) filters [30]. The SSIM values between $S(x_0)$ and each $S(x_t)$ are used to adaptively select an appropriate T to add noise, with the goal of removing tampered traces while preserving the overall structure of the input image.

Starting from the same noised image x_T, the two backward processes diverge: *unconditional denoising* produces a clean image without tampered traces, as the pre-trained diffusion model has learned a clean data distribution from untampered images. In contrast, *conditional denoising*

reconstructs the image with tampered traces, guided by the forged input and self-attention. SSIM scores are then calculated between the two denoised samples, $S(x_{c_t})$ and $S(x_{u_t})$, to determine the appropriate reverse timestep m for starting the aggregation of error maps, following the approach of [9]. The error map is computed using squared error.

5.2.1 Adaptive Number of Diffusion Timesteps Selection

The number of diffusion timestep T for adding noise plays a crucial role in our method. If T is too large, it would cause significant deviation from the input image, resulting in unexpected artifacts. Conversely, if T is too small, it cannot effectively remove the tampered areas, leaving manipulation traces in the unconditional reconstruction. Previous purification methods [90, 66, 82] select a fixed T for all images, which is clearly inappropriate. Inspired by the previous IML methods [5, 12, 123] that use high-pass filters to suppress image content, based on the idea that manipulation traces are more likely to be detected in the filtered results rather than in the image content. As discussed in the introduction and Fig. 5.1(a), high-frequency information is removed more quickly than image content in the diffusion forward process. Therefore, we use SRM filters [30] to process each x_t as $S(x_t)$, and the SSIM scores between each $S(x_t)$ and $S(x_0)$ are used to adaptively select an appropriate T. The basic idea is that when the SSIM using high-pass filters approaches 0, the SSIM without high-pass filters remains higher. This ensures that the forward process effectively removes manipulation traces while preserving the overall structure of the image. As illustrated in the examples in Fig. 5.2(a), the adaptively selected T causes the filtered image $S(x_T)$ to resemble random noise, while the image sample x_T still retains the overall structure. The reason for using SSIM is that it provides a clear cut-off value to indicate when two images are dissimilar (when the score is 0), whereas other metrics, such as mean square error (MSE), do not offer this property. We select 0.2 as the SSIM threshold to determine the appropriate T, meaning that when SSIM falls below 0.2, that timestep is selected as T.

5.2.2 Conditional Backward Process

Due to unexpected inconsistencies in authentic pixels when calculating the error directly from the unconditional reconstruction and the input, we modified the diffusion backward process into two branches, with the error now calculated between these two backward processes. By applying both conditional and unconditional backward processes to the same noised image x_T and ensuring that both use the same random noise at each timestep, we aimed to resolve these inconsistencies and improve localization performance.

For unconditional denoising, the process simply starts from x_T and gradually removes noise without any guidance, as described in Eq. (2.2). Our primary contribution lies in introducing conditional denoising without apply any external conditions, aiming to reconstruct an image that retains manipulation traces. This allows the inconsistency between the conditional and unconditional branches to effectively highlight the tampered regions.

Similarity guidance is employed to direct the reconstruction using the forged input, with the aim of guiding the model to reconstruct the image as closely as possible to the forged one, thereby preserving the tampered traces. Similar to [90, 82], we define the similarity metric as $D(\cdot)$ and the similarity guidance is given by:

$$\widetilde{\epsilon}(x_{c_t}, d, t) = \epsilon_\theta(x_{c_t}, t) - s_{d_t} \cdot \sigma_t \nabla_{x_{c_t}} D(x_t, x_{c_t}), \tag{5.1}$$

which is similar to classifier guidance shown in Eq. (2.4). The key difference is that the separate classifier is replaced by the similarity metric. Here, $\widetilde{\epsilon}(x_{c_t}, d, t)$ represents the conditional output guided by the similarity d. x_t and x_{c_t} are samples in forward and conditional backward process, respectively. s_{d_t} is the guidance scale that is proportional to added noise and it can be expressed as $s_{d_t} = s_d \cdot \sqrt{1-\bar{\alpha}_t}/\sqrt{\bar{\alpha}_t}$, Where s_d is a pre-defined initial guidance scale.

Self-attention Guidance: Using solely Eq. (5.1) for conditional reconstruction results in unsatisfactory localization outcomes because the similarity guidance is applied globally to the entire image, leading to false alarms in the untampered regions. To address this issue and focus the

reconstruction error more on the tampered region, we draw inspiration from [41], which demonstrates that the self-attention map from the diffusion U-Net overlaps with high-frequency details in the image. Since manipulation traces are most likely found in high-frequency regions, such as edge inconsistencies, we incorporate self-attention guidance into the conditional reconstruction. The self-attention in U-Net is implemented as multi-head self-attention [87], with the number of attention heads denoted by N. Let Q_t^h denote the query, K_t^h denote the key and V_t^h denote the value. The attention on the hth head at timestep t is:

$$A(Q_t^h, K_t^h, V_t^h) = \text{softmax}(Q_t^h (K_t^h)^T / \sqrt{d}) \cdot V_t^h. \tag{5.2}$$

The stacked self-attention maps across all attention heads at timestep t is $A_{s_t} \in \mathbb{R}^{N \times (HW) \times (HW)}$, where H and W denote the height and width, respectively. Then, A_{s_t} is processed by global average pooling (GAP), reshaping Reshape$(\cdot)$ and upsampling Upsample$(\cdot)$ to match the dimensions of image sample x_{c_t}. The final aggregated attention A_{c_t} from all attention heads at timestep t is:

$$A_{c_t} = \text{Upsample}(\text{Reshape}(\text{GAP}(A_{s_t}))). \tag{5.3}$$

As shown in Fig. 5.2(b), once we have the attention map, we can use the activated information to guide the generation, thus the reconstruction can focus more on these regions. The basic idea is to apply Gaussian blur only to the activated regions and then use the residual information between the blurred and unblurred image samples to guide the generation in a classifier-free manner. Let $M_{c_t}^i$ denote the binary mask value, and $A_{c_t}^i$ denote the self-attention map value at the ith pixel. Given an attention mask threshold τ, we first threshold A_{c_t} to a binary mask M_{c_t} using:

$$M_{c_t} = \begin{cases} M_{c_t}^i = 1, & \text{if } A_{c_t}^i > \tau, \\ M_{c_t}^i = 0, & \text{otherwise.} \end{cases} \tag{5.4}$$

For the Gaussian blur process, we follow the method outlined in [41] to generate blurred sam-

ples x_{b_t} from x_{c_t}. This approach helps mitigate the side effects of reducing Gaussian noise when applying Gaussian blur, as discussed in [41]. Finally, M_{c_t} is used to obtain the masked blurred samples $\hat{x}_{b_t}$, where only the regions with high activation in self-attention are blurred. The residual information is then used to guide the generation. Let $\odot$ denote element-wise multiplication, and $\tilde{\epsilon}(x_{c_t}, a, t)$ denote the guided output using self-attention guidance a, and s_f be the self-attention guidance scale. The masking and final self-attention guiding process is:

$$\hat{x}_{b_t} = (1 - M_{c_t}) \odot x_{c_t} + M_{c_t} \odot x_{b_t}, \tag{5.5}$$

$$\tilde{\epsilon}(x_{c_t}, a, t) = \epsilon_\theta(x_{c_t}, t) + s_f \cdot (\epsilon_\theta(x_{c_t}, t) - \epsilon_\theta(\hat{x}_{b_t}, t)). \tag{5.6}$$

This allows using the masked residual information to guide the generation, making the denoising process concentrate more on the masked region.

The complete conditional denoising process incorporates both similarity and self-attention guidance, applying guidance from both global and local perspectives. The final conditional generation output guided by both a and d is:

$$\begin{aligned} \tilde{\epsilon}(x_{c_t}, a, d, t) = \ & \epsilon_\theta(x_{c_t}, t) + s_f \cdot (\epsilon_\theta(x_{c_t}, t) \\ & - \epsilon_\theta(\hat{x}_{b_t}, t)) - s_{d_t} \cdot \sigma_t \nabla_{x_{c_t}} D(x_t, x_{c_t}). \end{aligned} \tag{5.7}$$

5.2.3 Error Map Aggregation

Unlike the forward process, where SSIM consistently decreases, in the backward process, SSIM first decreases and then increases. This behavior, also noted in [9] using external conditions, occurs because the unconditional branch initially reconstructs tampered information into a clean distribution, while the conditional branch works to reverse manipulation traces, leading to a decrease in SSIM. Once SSIM reaches its minimum, both branches have reconstructed the tampered regions and start to reconstruct the original information, causing SSIM to rise. Following [9], we aggregate error maps starting from the reverse timestep m (where SSIM is lowest). We calculate

the error map using the squared error formula: $(x_{c_t} - x_{u_t})^2$. The final aggregated error map is the average of all error maps from reverse timestep m to 0. The final localization map $E(x_u, x_c)$ is obtained by:

$$E(x_u, x_c) = \frac{\sum_{t=0}^{m}(x_{c_t} - x_{u_t})^2}{m+1}. \tag{5.8}$$

5.3 Experimental Results

We first present the experimental setup, including implementation details, datasets, and evaluation metrics. We then compare the IML performance of our method against State-of-The-Art approaches. Finally, we provide ablation studies.

5.3.1 Experimental Setup

Datasets: We use six IML datasets for evaluation: CASIAv1 [24], Colombia [42], Coverage [95], NIST16 [32], CIMD [121] and MagicBrush [112]. The first five datasets contain only standard manipulation types, which are splicing, copy-move, and removal. MagicBrush, however, is a novel instruction-guided manipulation dataset that features previously unseen and more complex tampered types, such as color changes, action changes, and object alterations. This dataset is closer to real-world manipulations and is particularly valuable for evaluating a model's generalizability. For the CIMD dataset, we applied the uncompressed subset, which is intended for evaluating image editing IML methods.

Evaluation metrics: We use two thresholding-agnostic metrics for evaluation: Area Under the Receiver Operating Characteristic curve (AUC) and Average Precision (AP). These two evaluation metrics do not require predefined thresholds, making the evaluation more fair. Some previous works [51, 18] have also used permuted metrics for evaluation, as they argue that it can sometimes be ambiguous to determine which segments should be identified as tampered. Accordingly, we also provide evaluation results using permuted metrics. To ensure fair evaluation across all methods, we used the same evaluation code for quantitative results.

Implementation details: Our method does not require training or external conditions. We

Method	CASIAv1		Columbia		Coverage		NIST16		CIMD		Average	
	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP
NOI1[60]	<u>0.586</u>	<u>0.140</u>	0.539	0.387	0.580	<u>0.168</u>	0.511	0.115	<u>0.680</u>	<u>0.060</u>	<u>0.579</u>	<u>0.174</u>
CFA1[29]	0.498	0.100	<u>0.641</u>	<u>0.445</u>	0.533	0.133	0.503	0.101	0.427	0.016	0.520	0.159
MCA [4]	0.542	0.117	0.513	0.270	0.536	0.124	0.520	0.083	0.521	0.020	0.526	0.123
NoisePrint [18]	0.514	0.091	0.563	0.359	0.515	0.123	0.450	0.114	0.543	0.018	0.517	0.141
IVC [17]	0.531	0.109	0.511	0.291	0.532	0.140	0.532	0.092	0.561	0.021	0.533	0.131
CFA2 [22]	0.531	0.104	0.530	0.411	0.524	0.143	0.480	0.095	0.510	0.017	0.515	0.154
NOI2 [59]	0.574	0.135	0.559	0.353	<u>0.598</u>	0.161	0.519	0.089	0.506	0.018	0.551	0.151
NOI4 [88]	0.535	0.106	0.536	0.313	0.537	0.130	0.494	0.085	0.634	0.043	0.547	0.135
BLK [53]	0.541	0.112	0.624	0.416	<u>0.598</u>	0.154	**0.583**	<u>0.136</u>	0.494	0.027	0.568	0.169
Ours	**0.587**	**0.162**	**0.682**	**0.461**	**0.622**	**0.208**	<u>0.556</u>	**0.160**	**0.690**	**0.068**	**0.627**	**0.212**

Table 5.1: Evaluation results of unsupervised methods for the Standard Manipulation task. Average scores are calculated across five datasets, with the best and second-best performances highlighted in bold and underlined.

used the pre-trained diffusion model from [21], which was trained on ImageNet [20]. The method is implemented using Pytorch [74] on an A40 GPU. For the diffusion model itself, we did not modify any of the diffusion settings except for the diffusion timestep T. For our proposed components, we set the initial similarity scale to $s_d = 10^4$, and the threshold for selecting the appropriate T is set to 0.2. In self-attention guidance, the guidance scale s_f is set to 1.3, the attention threshold τ is 1.3, and the blur sigma is 3.

Method	Training Data Size	MagicBrush	
		AUC	AP
Mantra-Net [99]	64K	0.426	0.156
PSCC-Net [56]	100K	0.375	0.140
CAT-Net [51]	858K	0.392	0.155
Hifi-Net [16]	1,710K	0.480	0.169
CR-CNN [102]	12.5K	0.515	0.193
MVSS-Net [12]	12.5K + NMA	**0.578**	**0.270**
WSCL [109]	12.5K	0.516	0.170
Ours	**None**	<u>0.543</u>	<u>0.206</u>

Table 5.2: Evaluation results for the Novel Manipulation task for both fully supervised and weakly supervised methods. NMA refers to Naive Manipulation Augmentation, which includes techniques such as cropping and pasting squared areas, and utilizing OpenCV inpainting functions [84, 6]. The best and second-best performances are highlighted in bold and underline, respectively.

5.3.2 Comparison with SoTA Methods

We conducted a comprehensive comparison with sixteen state-of-the-art (SoTA) methods, spanning unsupervised, weakly-supervised, and fully-supervised approaches. Crucially, all selected methods have open-source code, ensuring a fair evaluation. The unsupervised methods include [60, 59, 88, 29, 22, 53, 4, 17, 18], with the first six being implemented by MKLab [108]. For weakly-supervised methods, we evaluate [109]. The fully-supervised methods include [5, 51, 56, 99, 34, 12]. Using their open-source code, we generated localization maps and applied the same evaluation code to obtain quantitative results, maintaining consistency for a fair comparison. Abbreviations for each method follow those used in prior work.

Comparison using standard manipulation datasets: Table 5.1 provides evaluation results on five standard IML datasets for unsupervised methods. In most cases, our training-free method achieves the best localization performance across almost all datasets, except for the AUC score on the NIST16 dataset. Regarding the AUC performance on NIST16, our method does not outperform BLK, as all images in NIST16 are JPEG compressed, and BLK is specifically designed for JPEG format. Additionally, our method achieves significantly higher average performance than other approaches, demonstrating much stronger localization ability.

Comparison on the MagicBrush dataset: Table 5.2 shows the evaluation results comparing fully-supervised and weakly-supervised IML methods on MagicBrush [112], a recent IML dataset containing new tampered types. For methods trained on a dataset size of 12.5K, CASIAv2 [24] was used as the training set, while other methods used their own synthetic datasets. Table 5.2 shows results demonstrating that even fully-supervised methods trained on large datasets often struggle to adapt to new manipulation types, exhibiting low generalizability. In contrast, our training-free method delivers competitive results without any training images. Although MVSS-Net outperforms our method, it employs naive manipulation augmentation (NMA), such as cropping and pasting squared areas, and utilizing OpenCV inpainting functions [84, 6], thereby increasing its training data diversity beyond the 12.5K samples.

Comparison using Permuted Metrics: Some IML methods [51, 18] use permuted metrics

Method	CASIAv1		Columbia		Coverage		NIST16		CIMD		Average	
	p-AUC	p-AP	p-AUC	p-AP	p-AUC	p-AP	p-AUC	p-AP	p-AUC	p-AP	p-AUC	p-AP
NOI1 [60]	<u>0.684</u>	<u>0.174</u>	0.722	**0.503**	<u>0.668</u>	<u>0.201</u>	<u>0.694</u>	<u>0.190</u>	<u>0.721</u>	<u>0.062</u>	<u>0.698</u>	<u>0.226</u>
CFA1 [29]	0.622	0.129	0.714	0.490	0.618	0.152	0.613	0.140	0.614	0.023	0.636	0.187
MCA [4]	0.604	0.125	0.553	0.292	0.567	0.130	0.567	0.089	0.589	0.022	0.576	0.132
NoisePrint [18]	0.537	0.097	0.603	0.380	0.567	0.138	0.655	0.181	0.565	0.020	0.592	0.163
IVC [17]	0.579	0.114	0.536	0.301	0.559	0.143	0.576	0.096	0.570	0.025	0.564	0.136
CFA2 [22]	0.606	0.127	**0.760**	0.575	0.606	0.161	0.634	0.164	0.598	0.021	0.641	0.210
NOI2 [59]	0.629	0.151	0.597	0.367	0.628	0.167	0.535	0.090	0.507	0.018	0.579	0.159
NOI4 [88]	0.582	0.116	0.577	0.327	0.560	0.133	0.572	0.096	0.616	0.044	0.581	0.143
BLK [53]	0.630	0.145	0.684	0.451	0.648	0.169	0.688	0.164	0.658	0.042	0.662	0.194
Ours	**0.696**	**0.199**	<u>0.741</u>	<u>0.497</u>	**0.715**	**0.238**	**0.757**	**0.242**	**0.750**	**0.073**	**0.732**	**0.250**

Table 5.3: Evaluation results of unsupervised methods using permuted metrics for the Standard Manipulation task. The average scores are computed across the five datasets, with the best and second-best performances highlighted in bold and underlined, respectively.

due to the potential ambiguity in determining which of the two segments is tampered with. For instance, in the context of splicing tampering, when IML localizes a region, it remains unclear whether the detected region was spliced into the image or if an undetected region was spliced into the image. Consequently, it is difficult to conclusively classify the detected region as tampered. Let G and P represent the ground-truth mask and the predicted mask, respectively, and $\complement$ denote the flipping operation. The *permuted average precision (AP)* is defined as:

$$\text{p-AP}(G, P) = \max(\text{AP}(G, P), \text{AP}(G, P^{\complement})). \tag{5.9}$$

The *permuted AUC* can be calculated similarly. The results are summarized in Table 5.3 and Table 5.4.

For the standard datasets, our method exhibits a slight decrease in performance compared to the best scores on the Columbia dataset but still achieves the highest overall performance across all five IML datasets, demonstrating strong overall effectiveness. On the novel (unseen) IML datasets, while our AP is not the second-best, it is only 0.007 lower than the top performance. This demonstrates that our training-free method remains competitive even against fully-supervised methods trained on large labeled datasets.

Experimental Results for Each Manipulation Types: For evaluating performance across

Method	Training Data Size	MagicBrush	
		p-AUC	p-AP
Mantra-Net [99]	64K	0.595	0.192
PSCC-Net [56]	100K	0.692	0.243
CAT-Net [51]	858K	0.656	0.235
Hifi-Net [16]	1,710K	**0.714**	<u>0.275</u>
CR-CNN [102]	12.5K	0.609	0.213
MVSS-Net [12]	12.5K + NMA	0.643	**0.279**
WSCL [109]	12.5K	0.625	0.221
Ours	**None**	<u>0.696</u>	0.272

Table 5.4: Evaluation results on the Novel Manipulation task using permuted metrics for both fully-supervised and weakly-supervised methods. The best and second-best performances are highlighted in bold and underline, respectively.

Method	Copy-move		Removal		Splicing	
	AUC	AP	AUC	AP	AUC	AP
NOI1 [60]	0.728	0.066	0.586	0.027	0.724	0.086
CFA1 [29]	0.421	0.016	0.454	0.020	0.407	0.012
MCA [4]	0.535	0.021	0.492	0.023	0.535	0.018
NoisePrint [18]	0.547	0.018	0.536	0.020	0.545	0.016
IVC [17]	0.568	0.027	0.542	0.021	0.574	0.026
CFA2 [22]	0.531	0.018	0.487	0.019	0.512	0.014
NOI2 [59]	0.511	0.019	0.501	0.019	0.505	0.017
NOI4 [88]	0.676	0.047	0.556	0.023	0.670	0.059
BLK [53]	0.520	0.037	0.458	0.018	0.505	0.025
Ours	**0.735**	**0.067**	**0.588**	**0.032**	**0.754**	**0.108**

Table 5.5: AUC and AP performance comparison of various methods for the three manipulation types on the CIMD dataset [121].

different manipulation types, we utilized the CIMD dataset [121], which provides consistent tampered image samples for each manipulation type, ensuring for a fair assessment of performance. As shown in Table 5.5, our method outperforms other State-of-The-Art (SoTA) IML methods across all standard manipulation types.

Visualization: Fig. 5.3 presents the visualization of the IML results. Compared to other methods, our approach offers improved coverage of the tampered regions, despite not requiring any training. However, because our method is training-free and does not rely on datasets or pixel-level masks for supervision, it struggles to define the edges of the tampered regions precisely. We plan to address this limitation in our future work.

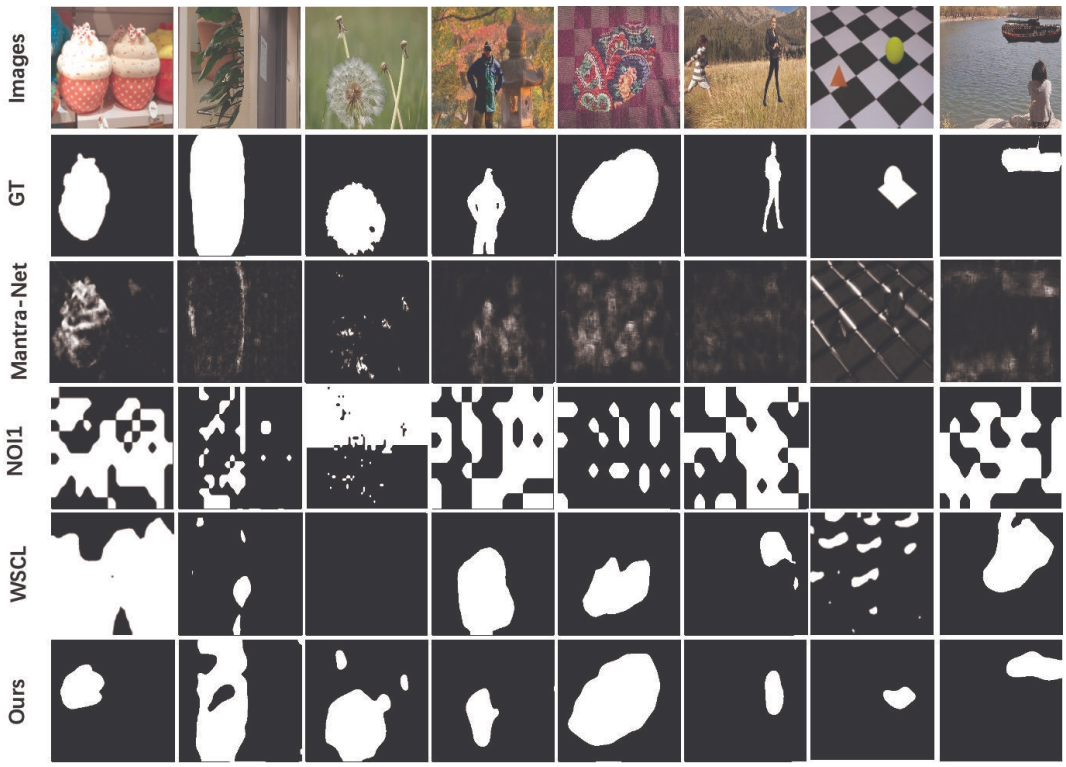

Figure 5.3: Visualization results are shown from top to bottom: the tampered images, ground-truth masks, results of the fully-supervised method Mantra-Net [99], the unsupervised method NOI1 [60], the weakly-supervised method WSCL [109], and our training-free method.

5.3.3 Ablation Study

We conduct ablation studies using CASIAv1 [24] to demonstrate the effectiveness of the proposed components.

Effectiveness of conditional guidance: We assess the impact of conditional guidance, focusing on similarity and self-attention guidance. As shown in Table 5.6, using only similarity guidance does not produce satisfactory results. This is because similarity guidance directs reconstruction globally, leading to unintended false alarms, as explained in the introduction and method sections. On the other hand, using only self-attention guidance significantly improves performance. The best results are achieved when both similarity and self-attention guidance are combined, underscoring the importance of both. In summary, similarity guidance increases the inconsistency between the two branches in the tampered area, while self-attention guidance focuses more on the

Unconditional	Similarity	Self-attention	AUC	AP
✓			0.544	0.102
✓	✓		0.514	0.109
✓		✓	0.573	0.127
✓	✓	✓	**0.587**	**0.162**

Table 5.6: Ablation study on different conditions.

T	$T = 10$	$T = 50$	$T = 100$	$T = 200$	Adap
AUC	0.532	0.577	0.558	0.467	**0.587**
AP	0.119	0.155	0.143	0.107	**0.162**

Table 5.7: IML performance comparisons using various fixed timesteps T and our adaptive T.

tampered area and reduces false positives. Both are essential for optimal performance.

Adaptive diffusion timestep selection: Our method adaptively selects an appropriate diffusion timesteps T to add noise to the input image, thereby avoiding the issue of T being too low or too high. As shown in Table 5.7, increasing T initially improves performance but eventually causes a decline. Although there might be an optimal fixed T, finding it would require extensive experimentation. In contrast, our adaptive approach achieves the best performance without the need for such experiments.

CHAPTER 6

Conclusion

This dissertation aims to advance research in image manipulation detection (IMD) by improving the generalizability of detection methods in real-world scenarios. It includes three contributions, first, it introduces a new benchmark dataset CIMD designed to evaluate existing methods under more challenging and realistic conditions, along with a fully supervised model that significantly improves detection performance without requiring any expansion of training data. Second, it presents a unified framework that operates in both weakly supervised and unsupervised settings, reducing dependence on pixel-level annotations. Third, it proposes a training-free paradigm based on diffusion models that requires neither model training nor external training datasets. This method detects manipulations by exploiting inconsistencies between conditional and unconditional diffusion-based reconstructions.

6.1 Summary of Contributions

6.1.1 Challenging Image Manipulation Detection Benchmark

In this work, we focus on image manipulation detection under particularly challenging cases, including tiny manipulations and double-compression with identical quality factors. We observe that many existing SoTA methods rely on strict assumptions about manipulated images that rarely hold in real-world settings. To enable more accurate evaluation under such conditions, we introduce a new high-quality benchmark dataset with precise annotations specifically tailored to these challenging scenarios. Alongside this dataset, we design a two-branch HRNet-based model that significantly outperforms prior approaches on both tiny-manipulation and same-QF double-compression tasks, all without requiring any enlargement of the training set.

6.1.2 Unified Unsupervised and Weakly Supervised Framework

In this work, we propose a unified framework that integrates unsupervised and weakly supervised learning under limited supervision. The framework leverages INR-based reconstruction errors as manipulation priors, employs selective pixel-level contrastive learning to restrict optimization to high-confidence regions, and incorporates adaptive pooling to produce more robust image-level predictions during training. In the unsupervised setting, we adopt a self-distillation strategy with pseudo-label selection to guide network optimization, where high-confidence image-level predictions are used to supervise intermediate feature representations. Extensive experiments demonstrate that this framework not only outperforms existing unsupervised and weakly supervised methods, but also achieves performance comparable to fully supervised approaches on novel (unseen) manipulation types.

6.1.3 Training-Free Diffusion-Based Method

In this work, we proposed a training-free method for image manipulation localization using pre-trained diffusion models. By adaptively selecting diffusion timesteps and comparing conditional and unconditional reconstructions, the method produces localization maps directly from reconstruction inconsistencies. This training-free approach achieves superior performance on standard manipulation tasks while exhibiting strong robustness to unseen manipulation types.

6.2 Limitations

Despite the contributions presented in the previous section, several limitations remain in this dissertation, which we leave for future work to address.

First, methods without pixel-level supervision cannot accurately localize manipulation boundaries. Technically, this limitation arises because weakly supervised, unsupervised, and training-free approaches never observe true pixel-level annotations during training. Without such ground-truth guidance, models lack explicit boundary constraints, making it inherently difficult to learn sharp and well-aligned manipulation contours.

Second, although our proposed CIMD benchmark enables evaluation of image manipulation detection under several challenging scenarios, its scale and manipulation diversity remain insufficient, as it currently includes only two types of challenging cases. CIMD was intentionally constructed as an evaluation benchmark to assess state-of-the-art image manipulation detection (IMD) methods under more challenging conditions. However, as a dataset designed primarily for evaluation, it does not yet fully capture the breadth and variability of manipulations encountered in real-world forensic scenarios.

Third, although the diffusion-based method is training-free—requiring neither model training nor external datasets and exhibiting strong adaptability to unseen manipulations—the computational cost and inference time remain non-trivial. In our pipeline, we must first determine the optimal diffusion timestep for the forward noising process, which is accomplished using high-pass filtering followed by similarity-based comparison. We then perform two reconstruction processes—one conditional and one unconditional—where the conditional branch incorporates similarity-guided signals in a classifier-guided manner as well as self-attention guidance in a classifier-free manner. Finally, multi-timestep reconstruction error maps are aggregated to generate the final localization output via the inconsistency map. This multi-stage workflow limits its suitability for real-time deployment.

In addition, because this training-free approach relies on diffusion-based reconstruction inconsistency, it becomes less effective when the input image is itself edited by a diffusion model. Diffusion-based outpainting and enhancement start from the same underlying generative distribution, which significantly reduces the reconstruction inconsistency signal. As a result, localization performance degrades for diffusion-edited images.

6.3 Future Work

Based on the limitations of the proposed methods, we can summarize several promising research directions for future work in image manipulation detection that may further enhance performance and improve generalizability. A first and highly practical direction is dataset expan-

sion. Although the proposed CIMD benchmark effectively evaluates IMD models on small-region manipulations and double-compression cases, its overall scale and diversity remain limited. Future work should focus on constructing larger and more comprehensive datasets that incorporate a broader range of environments, richer manipulation types, and more challenging scenarios—such as full-image tampering and images containing multiple manipulation types simultaneously.

A second direction is to broaden manipulation detection beyond still images. Real-world media are inherently multimodal—including video, images, audio, and text—and restricting forensic analysis to a single modality leaves substantial gaps in practical coverage, since real-world cases rarely involve only one form of media. Therefore, developing multimodal manipulation detection tools is crucial.

Third, we can improve the inference speed of diffusion-based methods. The current slow runtime and high computational overhead make these approaches impractical for real-world deployment, where fast and reliable responses are required. Also the future diffusion based method should be able to detect the diffusion editting image as well.

With the rapid development of vision–language models (VLMs) such as [3], future IMD systems can benefit from incorporating these models to detect manipulations in a more semantic, interpretable, and interactive manner. VLMs enable models not only to identify pixel-level inconsistencies but also to reason about what was manipulated and why it is semantically suspicious. This opens the door to more transparent explanations of manipulation evidence. Beyond improving interpretability, an end-to-end VLM-driven pipeline could support interactive image forensics—allowing a user to upload an image and directly ask manipulation-related questions (e.g., “Is the background manipulated?”, “Is the object’s color altered?”, “Is any part of the image inconsistent with the rest?”). Such capabilities would greatly enhance usability and strengthen the practical value of IMD systems in real-world applications.

BIBLIOGRAPHY

[1] Irene Amerini, Lamberto Ballan, Roberto Caldelli, Alberto Del Bimbo, and Giuseppe Serra, *A sift-based forensic method for copy–move attack detection and transformation recovery*, IEEE transactions on information forensics and security (2011).

[2] Tomer Amit, Tal Shaharbany, Eliya Nachmani, and Lior Wolf, *Segdiff: Image segmentation with diffusion probabilistic models*, arXiv preprint arXiv:2112.00390 (2021).

[3] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al., *Qwen2. 5-vl technical report*, arXiv preprint arXiv:2502.13923 (2025).

[4] Quentin Bammey, Rafael Grompone von Gioi, and Jean-Michel Morel, *An adaptive neural network for unsupervised mosaic consistency analysis in image forensics*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 14194–14204.

[5] Belhassen Bayar and Matthew C Stamm, *Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection.*, IEEE Transactions on Information Forensics and Security **13** (2018), no. 11, 2691–2706.

[6] Marcelo Bertalmio, Andrea L Bertozzi, and Guillermo Sapiro, *Navier-stokes, fluid dynamics, and image and video inpainting*, Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, vol. 1, IEEE, 2001, pp. I–I.

[7] Xiuli Bi, Yang Wei, Bin Xiao, and Weisheng Li, *Rru-net: The ringed residual u-net for image splicing forgery detection.*, IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2019.

[8] Luca Bondi, Silvia Lameri, David Güera, Paolo Bestagini, Edward J Delp, and Stefano Tubaro, *Tampering detection and localization through clustering of camera-based cnn features*, 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE, 2017, pp. 1855–1864.

[9] Yiming Che, Fazle Rafsani, Jay Shah, Md Mahfuzur Rahman Siddiquee, and Teresa Wu, *Anofpdm: Anomaly segmentation with forward process of diffusion models for brain mri*, arXiv preprint arXiv:2404.15683 (2024).

[10] Kai Chen, Lanqing Hong, Hang Xu, Zhenguo Li, and Dit-Yan Yeung, *Multisiam: Self-supervised multi-instance siamese representation learning for autonomous driving*, Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 7546–7554.

[11] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam, *Rethinking atrous convolution for semantic image segmentation*, arXiv preprint arXiv:1706.05587 (2017).

[12] Xinru Chen, Chengbo Dong, Jiaqi Ji, Juan Cao, and Xirong Li, *Image manipulation detection by multi-view multiscale supervision.*, IEEE/CVF International Conference on Computer Vision, 2021, pp. 14185–14193.

[13] Yinbo Chen, Sifei Liu, and Xiaolong Wang, *Learning continuous image representation with local implicit image function*, Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 8628–8638.

[14] Yuwei Chen, Ming-Ching Chang, Mattias Kirchner, Zhenfei Zhang, Xin Li, Arslan Basharat, and Anthony Hoogs, *A semantically impactful image manipulation dataset: Characterizing image manipulations using semantic significance*, 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), IEEE, 2025, pp. 7659–7668.

[15] Zeyuan Chen, Yinbo Chen, Jingwen Liu, Xingqian Xu, Vidit Goel, Zhangyang Wang, Humphrey Shi, and Xiaolong Wang, *Videoinr: Learning video implicit neural representa-*

tion for continuous space-time super-resolution, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 2047–2057.

[16] Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S Huang, and Lei Zhang, *Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation*, IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 5386–5395.

[17] Chang-Hee Choi, Jung-Ho Choi, and Heung-Kyu Lee, *Cfa pattern identification of digital cameras using intermediate value counting*, Proceedings of the thirteenth ACM multimedia workshop on Multimedia and security, 2011, pp. 21–26.

[18] Davide Cozzolino and Luisa Verdoliva, *Noiseprint: A cnn based camera model fingerprint.*, IEEE Transactions on Information Forensics and Security **15** (2019), 144–159.

[19] Maxime Daisy, Pierre Buyssens, David Tschumperlé, and Olivier Lézoray, *A smarter exemplar-based inpainting algorithm using local and global heuristics for more geometric coherence.*, IEEE International Conference on Image Processing (ICIP), IEEE, 2014, pp. 4622–4626.

[20] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, *Imagenet: A large-scale hierarchical image database*, 2009 IEEE conference on computer vision and pattern recognition, Ieee, 2009, pp. 248–255.

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

[22] Ahmet Emir Dirik and Nasir Memon, *Image tamper detection based on demosaicing artifacts*, 2009 16th IEEE International Conference on Image Processing (ICIP), IEEE, 2009, pp. 1497–1500.

[23] J. Dong, W. Wang, and T. Tan, *CASIA image tampering detection evaluation database*, `http://forensics.idealtest.org`, 2010.

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

[25] Emilien Dupont, Adam Goliński, Milad Alizadeh, Yee Whye Teh, and Arnaud Doucet, *Coin: Compression with implicit neural representations*, arXiv preprint arXiv:2103.03123 (2021).

[26] Tolga Ergen and Suleyman Serdar Kozat, *Unsupervised anomaly detection with lstm neural networks*, IEEE transactions on neural networks and learning systems **31** (2019), no. 8, 3127–3141.

[27] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al., *A density-based algorithm for discovering clusters in large spatial databases with noise*, kdd, vol. 96, 1996, pp. 226–231.

[28] Yutong Feng, Yifan Feng, Haoxuan You, Xibin Zhao, and Yue Gao, *Meshnet: Mesh neural network for 3d shape representation*, Proceedings of the AAAI conference on artificial intelligence, vol. 33, 2019, pp. 8279–8286.

[29] Pasquale Ferrara, Tiziano Bianchi, Alessia De Rosa, and Alessandro Piva, *Image forgery localization via fine-grained analysis of cfa artifacts*, IEEE Transactions on Information Forensics and Security **7** (2012), no. 5, 1566–1577.

[30] Jessica Fridrich and Jan Kodovsky, *Rich models for steganalysis of digital images*, IEEE Transactions on information Forensics and Security **7** (2012), no. 3, 868–882.

[31] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, *Generative adversarial networks*, Communications of the ACM **63** (2020), no. 11, 139–144.

[32] 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., IEEE/CVF Winter Conference on Applications of Computer Vision Workshops, IEEE, 2019, pp. 63–72.

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

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

[35] Raia Hadsell, Sumit Chopra, and Yann LeCun, *Dimensionality reduction by learning an invariant mapping*, 2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06), vol. 2, IEEE, 2006, pp. 1735–1742.

[36] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick, *Momentum contrast for unsupervised visual representation learning*, Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 9729–9738.

[37] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, *Deep residual learning for image recognition*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[38] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio, *Learning deep representations by mutual information estimation and maximization*, arXiv preprint arXiv:1808.06670 (2018).

[39] Jonathan Ho, Ajay Jain, and Pieter Abbeel, *Denoising diffusion probabilistic models*, Advances in neural information processing systems **33** (2020), 6840–6851.

[40] Jonathan Ho and Tim Salimans, *Classifier-free diffusion guidance*, arXiv preprint arXiv:2207.12598 (2022).

[41] Susung Hong, Gyuseong Lee, Wooseok Jang, and Seungryong Kim, *Improving sample quality of diffusion models using self-attention guidance*, Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 7462–7471.

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

[43] Jie Hu, Li Shen, and Gang Sun, *Squeeze-and-excitation networks*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7132–7141.

[44] Juan Hu, Xin Liao, Wei Wang, and Zheng Qin, *Detecting compressed deepfake videos in social networks using frame-temporality two-stream convolutional network*, IEEE Transactions on Circuits and Systems for Video Technology **32** (2021), no. 3, 1089–1102.

[45] Xuefeng Hu, Zhihan Zhang, Zhenye Jiang, Syomantak Chaudhuri, Zhenheng Yang, and Ram Nevatia, *Span: Spatial pyramid attention network for image manipulation localization.*, European Conference on Computer Vision (ECCV),, Springer, 2020, pp. 312–328.

[46] Fangjun Huang, Jiwu Huang, and Yun Qing Shi, *Detecting double jpeg compression with the same quantization matrix*, IEEE Transactions on Information Forensics and Security **5** (2010), no. 4, 848–856.

[47] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani, *Imagic: Text-based real image editing with diffusion models*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 6007–6017.

[48] Diederik P Kingma and Max Welling, *Auto-encoding variational bayes*, arXiv preprint arXiv:1312.6114 (2013).

[49] Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al., *Siamese neural networks for one-shot image recognition*, ICML deep learning workshop, vol. 2, Lille, 2015.

[50] Ho Man Kwan, Ge Gao, Fan Zhang, Andrew Gower, and David Bull, *Hinerv: Video compression with hierarchical encoding-based neural representation*, Advances in Neural Information Processing Systems **36** (2024).

[51] Myung-Joon Kwon, Seung-Hun Nam, In-Jae Yu, Heung-Kyu Lee, and Changick Kim, *Learning jpeg compression artifacts for image manipulation detection and localization*, International Journal of Computer Vision (2022), 1875–1895.

[52] Shikun Li, Xiaobo Xia, Shiming Ge, and Tongliang Liu, *Selective-supervised contrastive learning with noisy labels*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 316–325.

[53] Weihai Li, Yuan Yuan, and Nenghai Yu, *Passive detection of doctored jpeg image via block artifact grid extraction*, Signal Processing **89** (2009), no. 9, 1821–1829.

[54] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick, *Microsoft coco: Common objects in context*, European conference on computer vision (ECCV), 2014, pp. 740–755.

[55] Dongju Liu and Jian Yu, *Otsu method and k-means*, 2009 Ninth International conference on hybrid intelligent systems, vol. 1, IEEE, 2009, pp. 344–349.

[56] 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** (2022), no. 11, 7505–7517.

[57] Ilya Loshchilov and Frank Hutter, *Decoupled weight decay regularization*, arXiv preprint arXiv:1711.05101 (2017).

[58] Xiankai Lu, Wenguan Wang, Chao Ma, Jianbing Shen, Ling Shao, and Fatih Porikli, *See more, know more: Unsupervised video object segmentation with co-attention siamese networks*, Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 3623–3632.

[59] Siwei Lyu, Xunyu Pan, and Xing Zhang, *Exposing region splicing forgeries with blind local noise estimation*, International journal of computer vision **110** (2014), 202–221.

[60] Babak Mahdian and Stanislav Saic, *Using noise inconsistencies for blind image forensics*, Image and vision computing **27** (2009), no. 10, 1497–1503.

[61] Gaël Mahfoudi, Badr Tajini, Florent Retraint, Frederic Morain-Nicolier, Jean Luc Dugelay, and PIC Marc, *Defacto: image and face manipulation dataset*, 2019 27Th european signal processing conference (EUSIPCO), IEEE, 2019.

[62] Hannes Mareen, Dante Vanden Bussche, Fabrizio Guillaro, Davide Cozzolino, Glenn Van Wallendael, Peter Lambert, and Luisa Verdoliva, *Comprint: Image forgery detection and localization using compression fingerprints*, International Conference on Pattern Recognition (ICPR), 2022.

[63] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng, *Nerf: Representing scenes as neural radiance fields for view synthesis*, Communications of the ACM **65** (2021), no. 1, 99–106.

[64] Amirali Molaei, Amirhossein Aminimehr, Armin Tavakoli, Amirhossein Kazerouni, Bobby Azad, Reza Azad, and Dorit Merhof, *Implicit neural representation in medical imaging: A comparative survey*, Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 2381–2391.

[65] Tian-Tsong Ng, Jessie Hsu, and Shih-Fu Chang, *Columbia image splicing detection evaluation dataset*, DVMM lab. Columbia Univ CalPhotos Digit Libr (2009).

[66] Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar, *Diffusion models for adversarial purification*, arXiv preprint arXiv:2205.07460 (2022).

[67] Yakun Niu, Xiaolong Li, Yao Zhao, and Rongrong Ni, *Detection of double jpeg compression with the same quantization matrix via convergence analysis*, IEEE Transactions on Circuits and Systems for Video Technology **32** (2021), no. 5, 3279–3290.

[68] Yakun Niu, Benedetta Tondi, Yao Zhao, Rongrong Ni, and Mauro Barni, *Image splicing detection, localization and attribution via jpeg primary quantization matrix estimation and clustering*, IEEE Transactions on Information Forensics and Security **16** (2021), 5397–5412.

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

[70] Aaron van den Oord, Yazhe Li, and Oriol Vinyals, *Representation learning with contrastive predictive coding*, arXiv preprint arXiv:1807.03748 (2018).

[71] Nobuyuki Otsu, *A threshold selection method from gray-level histograms*, IEEE transactions on systems, man, and cybernetics **9** (1979), no. 1, 62–66.

[72] Xunyu Pan, Xing Zhang, and Siwei Lyu, *Exposing image forgery with blind noise estimation*, Proceedings of the thirteenth ACM multimedia workshop on Multimedia and security, 2011, pp. 15–20.

[73] Jinseok Park, Donghyeon Cho, Wonhyuk Ahn, and Heung-Kyu Lee, *Double jpeg detection in mixed jpeg quality factors using deep convolutional neural network*, European conference on computer vision (ECCV), 2018, pp. 636–652.

[74] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al., *Pytorch: An imperative style, high-performance deep learning library*, Advances in neural information processing systems **32** (2019).

[75] Deepak Pathak, Evan Shelhamer, Jonathan Long, and Trevor Darrell, *Fully convolutional multi-class multiple instance learning*, arXiv preprint arXiv:1412.7144 (2014).

[76] Peng Peng, Tanfeng Sun, Xinghao Jiang, Ke Xu, Bin Li, and Yunqing Shi, *Detection of double jpeg compression with the same quantization matrix based on convolutional neural*

networks, Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), IEEE, 2018, pp. 717–721.

[77] Stanislav Pyatykh, Jürgen Hesser, and Lei Zheng, *Image noise level estimation by principal component analysis*, IEEE transactions on image processing **22** (2012), no. 2, 687–699.

[78] Tingting Qiao, Jing Zhang, Duanqing Xu, and Dacheng Tao, *Mirrorgan: Learning text-to-image generation by redescription*, Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 1505–1514.

[79] Filip Radenović, Giorgos Tolias, and Ondřej Chum, *Fine-tuning cnn image retrieval with no human annotation*, IEEE transactions on pattern analysis and machine intelligence **41** (2018), no. 7, 1655–1668.

[80] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, *U-net: Convolutional networks for biomedical image segmentation*, Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18, Springer, 2015, pp. 234–241.

[81] Jing Shi, Ning Xu, Trung Bui, Franck Dernoncourt, Zheng Wen, and Chenliang Xu, *A benchmark and baseline for language-driven image editing*, Proceedings of the Asian Conference on Computer Vision, 2020.

[82] Matías Tailanián, Marina Gardella, Alvaro Pardo, and Pablo Musé, *Diffusion models meet image counter-forensics*, Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2024, pp. 3925–3935.

[83] Chenxin Tao, Xizhou Zhu, Weijie Su, Gao Huang, Bin Li, Jie Zhou, Yu Qiao, Xiaogang Wang, and Jifeng Dai, *Siamese image modeling for self-supervised vision representation learning*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 2132–2141.

[84] Alexandru Telea, *An image inpainting technique based on the fast marching method*, Journal of graphics tools **9** (2004), no. 1, 23–34.

[85] Ruben Tolosana, Ruben Vera-Rodriguez, Julian Fierrez, Aythami Morales, and Javier Ortega-Garcia, *Deepfakes and beyond: A survey of face manipulation and fake detection*, Information Fusion **64** (2020), 131–148.

[86] Dijana Tralic, Ivan Zupancic, Sonja Grgic, and Mislav Grgic, *Comofod—new database for copy-move forgery detection*, Proceedings ELMAR-2013, IEEE, 2013.

[87] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, *Attention is all you need*, Advances in neural information processing systems (2017).

[88] J. Wagner, *Noise analysis for image forensics*, April 2015, Online.

[89] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al., *Deep high-resolution representation learning for visual recognition*, IEEE transactions on pattern analysis and machine intelligence **43** (2020), no. 10, 3349–3364.

[90] Jinyi Wang, Zhaoyang Lyu, Dahua Lin, Bo Dai, and Hongfei Fu, *Guided diffusion model for adversarial purification*, arXiv preprint arXiv:2205.14969 (2022).

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

[92] Lijun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan, *Learning to detect salient objects with image-level supervision*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 136–145.

[93] Xiaolong Wang and Abhinav Gupta, *Unsupervised learning of visual representations using videos*, Proceedings of the IEEE international conference on computer vision, 2015, pp. 2794–2802.

[94] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli, *Image quality assessment: from error visibility to structural similarity*, IEEE transactions on image processing **13** (2004), no. 4, 600–612.

[95] Bihan Wen, Ye Zhu, Ramanathan Subramanian, Tian-Tsong Ng, Xuanjing Shen, and Stefan Winkler, *Coverage – a novel database for copy-move forgery detection*, IEEE International Conference on Image processing (ICIP), 2016.

[96] Julia Wolleb, Robin Sandkühler, Florentin Bieder, Philippe Valmaggia, and Philippe C Cattin, *Diffusion models for implicit image segmentation ensembles*, International Conference on Medical Imaging with Deep Learning, PMLR, 2022, pp. 1336–1348.

[97] Haiwei Wu, Yiming Chen, and Jiantao Zhou, *Rethinking image forgery detection via contrastive learning and unsupervised clustering*, arXiv preprint arXiv:2308.09307 (2023).

[98] Haiwei Wu, Jiantao Zhou, Jinyu Tian, and Jun Liu, *Robust image forgery detection over online social network shared images.*, IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 13440–13449.

[99] Yue Wu, Wael AbdAlmageed, and Premkumar Natarajan, *Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features.*, IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 9543–9552.

[100] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin, *Unsupervised feature learning via non-parametric instance discrimination*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 3733–3742.

[101] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He, *Attngan: Fine-grained text to image generation with attentional generative*

adversarial networks, Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 1316–1324.

[102] Chao Yang, Huizhou Li, Fangting Lin, Bin Jiang, and Hao Zhao, *Constrained r-cnn: A general image manipulation detection model.*, IEEE International Conference on Multimedia and Expo (ICME), IEEE, 2020, pp. 1–6.

[103] Jianquan Yang, Jin Xie, Guopu Zhu, Sam Kwong, and Yun-Qing Shi, *An effective method for detecting double jpeg compression with the same quantization matrix*, IEEE Transactions on Information Forensics and Security **9** (2014), no. 11, 1933–1942.

[104] Min Yang, Dongliang He, Miao Fan, Baorong Shi, Xuetong Xue, Fu Li, Errui Ding, and Jizhou Huang, *Dolg: Single-stage image retrieval with deep orthogonal fusion of local and global features*, IEEE/CVF International conference on Computer Vision, 2021, pp. 11772–11781.

[105] Shuzhou Yang, Moxuan Ding, Yanmin Wu, Zihan Li, and Jian Zhang, *Implicit neural representation for cooperative low-light image enhancement*, Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 12918–12927.

[106] Yijun Yang, Huazhu Fu, Angelica I Aviles-Rivero, Carola-Bibiane Schönlieb, and Lei Zhu, *Diffmic: Dual-guidance diffusion network for medical image classification*, International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2023, pp. 95–105.

[107] Yassine Yousfi and Jessica Fridrich, *An intriguing struggle of cnns in jpeg steganalysis and the onehot solution*, IEEE Signal Processing Letters **27** (2020), 830–834.

[108] Markos Zampoglou, Symeon Papadopoulos, and Yiannis Kompatsiaris, *Large-scale evaluation of splicing localization algorithms for web images*, Multimedia Tools and Applications **76** (2017), no. 4, 4801–4834.

[109] Yuanhao Zhai, Tianyu Luan, David Doermann, and Junsong Yuan, *Towards generic image manipulation detection with weakly-supervised self-consistency learning*, Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 22390–22400.

[110] Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka, *3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models*, arXiv preprint arXiv:2301.11445 (2023).

[111] Hang Zhang, Rongguang Wang, Jinwei Zhang, Chao Li, Gufeng Yang, Pascal Spincemaille, Thanh Nguyen, and Yi Wang, *Nerd: Neural representation of distribution for medical image segmentation*, arXiv preprint arXiv:2103.04020 (2021).

[112] Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su, *Magicbrush: A manually annotated dataset for instruction-guided image editing*, Advances in Neural Information Processing Systems **36** (2024).

[113] Linfeng Zhang, Chenglong Bao, and Kaisheng Ma, *Self-distillation: Towards efficient and compact neural networks*, IEEE Transactions on Pattern Analysis and Machine Intelligence **44** (2021), no. 8, 4388–4403.

[114] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala, *Adding conditional control to text-to-image diffusion models*, Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 3836–3847.

[115] Wenwei Zhang, Jiangmiao Pang, Kai Chen, and Chen Change Loy, *Dense siamese network for dense unsupervised learning*, European Conference on Computer Vision, Springer, 2022, pp. 464–480.

[116] Zhenfei Zhang and Tien D Bui, *Attention-based selection strategy for weakly supervised object localization*, 2020 25th International Conference on Pattern Recognition (ICPR), IEEE, 2021, pp. 10305–10311.

[117] Zhenfei Zhang and Ming-Ching Chang, *Two-stage dual augmentation with clip for improved text-to-sketch synthesis*, 2023 IEEE 6th International Conference on Multimedia Information Processing and Retrieval (MIPR), IEEE, 2023, pp. 1–6.

[118] Zhenfei Zhang, Ming-Ching Chang, and Tien D Bui, *Improving class activation map for weakly supervised object localization*, ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2022, pp. 2624–2628.

[119] Zhenfei Zhang, Ming-Ching Chang, and Xin Li, *Training-free image manipulation localization using diffusion models*, Proceedings of the AAAI Conference on Artificial Intelligence, vol. 39, 2025, pp. 10376–10384.

[120] Zhenfei Zhang, Tsung-Wei Huang, Guan-Ming Su, Ming-Ching Chang, and Xin Li, *Text-driven synchronized diffusion video and audio talking head generation*, 2024 IEEE 7th International Conference on Multimedia Information Processing and Retrieval (MIPR), IEEE, 2024, pp. 61–67.

[121] Zhenfei Zhang, Mingyang Li, and Ming-Ching Chang, *A new benchmark and model for challenging image manipulation detection*, Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, 2024, pp. 7405–7413.

[122] Zhenfei Zhang, Mingyang Li, Xin Li, Ming-Ching Chang, and Jun-Wei Hsieh, *Image manipulation detection with implicit neural representation and limited supervision*, European Conference on Computer Vision, Springer, 2024, pp. 255–273.

[123] Peng Zhou, Xintong Han, Vlad I Morariu, and Larry S Davis, *Learning rich features for image manipulation detection*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 1053–1061.